AD NUMBER
AD877852
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; MAY 1970. Other requests shall be referred to Air Force Armament Lab., Eglin AFB, FL.
AUTHORITY
AFATL ltr, 19 Apr 1976

(2)

AD877852

AFATL-TR-70-40

AD No. ___

DDC FILE COPY

USER'S MANUAL: EXTENDED CAPABILITY MAGNUS ROTOR AND BALLISTIC BODY 6-DOF TRAJECTORY PROGRAM

ALPHA RESEARCH, INC.

TECHNICAL REPORT AFATL-TR-70-40

MAY 1970

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Air Force Armament Laboratory (ATRA), Eglin AFB, Florida 32542.

AIR FORCE ARMAMENT LABORATORY

AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE

EGLIN AIR FORCE BASE, FLORIDA

113

USER'S MANUAL:

EXTENDED CAPABILITY MAGNUS
ROTOR AND BALLISTIC BODY
6-DOF TRAJECTORY PROGRAM

James E. Brunk

FOREWORD

The extended Capability Magnus Rotor and Ballistic Body 6-DOF Trajectory Program has been prepared under Contract No. F08635-70-C-0012 with the Air Force Armament Laboratory, Eglin Air Force Base, Florida, by Alpha Research, Inc., Santa Barbara, California. The programmer at Alpha Research, Inc. was Mr. William Davidson. The program monitor for the Armament Laboratory was Mr. Edward S. Sears (ATRA). This effort was conducted during the period 6 October 1969 to 6 April 1970.

This program is a modification of several earlier computer programs prepared for magnus-rotor flight dynamics investigations. The original program was prepared for the U. S. Army Edgewood Arsenal under Contract No. DA-18-108-AMC-236(A). The original program was later adapted to the Air Force Armament Laboratory computer facilities by Air Force personnel, and was used by Alpha Research, Inc. in conjunction with Air Force Contracts Nos. F08635-67-C-0135 and F08635-69-C-0106.

The present modified program is written in General Fortran IV language compatible with third generation computers such as the GE 635, IBM 360, and CDC 6400.

Information in this report is embargoed under the Department of State International Traffic In Arms Regulations. This report may be released to foreign governments by departments or agencies of the U. S. Government subject to approval of the Air Force Armament Laboratory (ATRA), Eglin AFB, Florida 32542, or higher authority within the Department of the Air Force. Private individuals or firms require a Department of State export license.

This technical report has been reviewed and is approved.

Charles K Arpke

CHARLES K. ARPKE, Lt Colonel, USAF
Chief, Technology Division

ABSTRACT

A six-degrees-of-freedom trajectory program for quasi-symmetric rigid bodies is described. The equations of motion are developed such that either a body-fixed or fixed-plane moving coordinate system can be utilized. Provision is made for large angle and angular rate motions, such as are experienced by magnus rotor munitions.

The aerodynamic system permits the usual aeroballistic coefficients to be expressed as tabular functions of angle of attack and Mach number; in addition, the magnus force, magnus moment, and rolling moment coefficients can be tabular functions of the nondimensional spin parameter, pd/2V. Additional aerodynamic terms are provided to account for body-fixed aerodynamic asymmetries and/or control inputs, aerodynamic roll angle effects, flow asymmetry with respect to the angle of attack plane at zero spin, and lateral c.g. offset.

The computer program is written in General Fortran IV language compatible with CDC 6400, IBM 360, and GE 635 data processing machines. Included in the report are the program input and output formats, flow charts of the main program subroutines, and a complete program listing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

(The reverse of this page is blank)

LIST OF ABBREVIATIONS AND SYMBOLS

Symbol	Definition	Fortran Equivalent
C_x	Axial force coefficient	CX
C_N	Normal force coefficient	CN
C_M	Overturning moment coefficient	CM
C_{m_q}	Damping derivative (α plane)	CMQ
C_{n_r}	Damping derivative (Magnus plane)	CNR
$C_{m_{p_r}}$	Angular velocity coupling derivative	CMPR
$C_{n_{p_q}}$	Angular velocity coupling derivative	CNPQ
C_{N_p}	Magnus force coefficient	CNPA
C_{ℓ}	Spin torque coefficient	CL
C_{ℓ_p}	Spin damping coefficient	CLP
C_{M_p}	Magnus moment coefficient	CMPA
C_{y_o}	Trim force coefficient along y body-fixed axis	CYO
C_{z_o}	Trim force coefficient along z body-fixed axis	CZO
C_{m_o}	Trim moment coefficient about y body-fixed axis	CMO
C_{n_o}	Trim moment coefficient about z body-fixed axis	CNO
C_{SF_1}	Side force coefficient due to aerodynamic roll angle	CSF1
C_{N_1}	Normal force coefficient due to aerodynamic roll angle	CN1
C_{SF_3}	Side force coefficient due to asymmetric vortices	CSF3
C_{SM_1}	Side moment coefficient due to aerodynamic roll angle.	CSM1
C_{M_1}	Pitching moment coefficient due to aerodynamic roll angle.	CM1
C_{SM_3}	Side moment coefficient due to asymmetric vortices	CSM3
$C_{\ell_{\Phi 1}}$	Roll moment coefficient due to aerodynamic roll angle,	CLPM1
$C_{\ell_{\Phi 2}}$	Roll moment coefficient due to aerodynamic roll angle,	CLPM2

LIST OF ABBREVIATIONS AND SYMBOLS (CONTINUED)

Symbol	Definition	Units	Fortran Equivalent
a	Velocity of sound	ft/sec	VOS
d	Aerodynamic reference length (body diameter)	ft	DEE
g	Acceleration due to gravity	ft/sec^2	G
I_x	Axial moment of inertia	slug-ft^2	DIX
I_y	Transverse moment of inertia about y axis	slug-ft^2	DI
$I_y = I$	Transverse moment of inertia (fixed-plane axes)	slug-ft^2	DI
I_z	Transverse moment of inertia about z axis	slug-ft^2	DIZ
I_{xy}	Product of inertia	slug-ft^2	DIXY
m	Mass	slugs	DMM
M	Mach number		EM
p	Spin rate, angular velocity about x axis	rad/sec	P, Y(4)
pd/2V	Non-dimensional spin parameter		PDV
q	Angular velocity about y axis	rad/sec	Q, Y(5)
r	Angular velocity about z axis	rad/sec	R,Y(6)
S	Aerodynamic reference area	ft^2	S
t	Time	sec	TIME
Δt	Time increment used for integration	sec sec	TSTEP, TNEW
u	Axial velocity in direction of x axis	ft/sec	U, Y(1)
u_A	Aerodynamic velocity in direction of x axis	ft/sec	VA 1
v	Velocity in direction of y axis	ft/sec	V, Y(2)
v_A	Aerodynamic velocity in direction of y axis	ft/sec	VA 2
V	Total velocity	ft/sec	CAPV
V_A	Total aerodynamic velocity	ft/sec	CAPVA

LIST OF ABBREVIATIONS AND SYMBOLS (CONTINUED)

Symbol	Definition	Units	Fortran Equivalent
w	Velocity in direction of z axis	ft/sec	W, Y(3)
w_A	Aerodynamic velocity in direction of z axis	ft/sec	VA 3
X	Horizontal coordinate	ft	X, Y(7)
$\dot{X}$	Velocity in direction of X coordinate	ft/sec	XDOT
$\dot{X}_w$	Wind velocity in direction of X coordinate	ft/sec	WDX
Δy	c. g. lateral offset from axis of symmetry	ft	DY
Y	Horizontal coordinate	ft	Y, Y(8)
$\dot{Y}$	Velocity in direction of Y coordinate	ft	YDOT
$\dot{Y}_w$	Wind velocity in direction of Y coordinate	ft	WDY
Z	Vertical coordinate	ft	Z, Y(9)
$\dot{Z}$	Velocity in direction of Z coordinate	ft	ZDOT
$\tilde{\alpha}$	Total angle of attack	radians degrees	ALPHA ALD
$\zeta_{1,2}$	Orientation of fins and wings, respectively	degrees radians	ZETD 1, 2 ZET 1, 2
$\eta_{1,2}$	Number of fins and wings, respectively		ETA 1, 2
θ	Euler angle	degrees	THETA
$\dot{\theta}$	Euler angle rate	rad/sec	THD
λ_0	Quaternion		Y(10)
λ_1	Quaternion		Y(11)
λ_2	Quaternion		Y(12)
λ_3	Quaternion		Y(13)
ζ	Orientation of cross velocity	radians	ZETA
ρ	Air density	slug/ft^3	RHO
φ	Euler angle	degrees	PHI
$\dot{\varphi}$	Euler angle rate	rad/sec	PHD

LIST OF ABBREVIATIONS AND SYMBOLS (CONCLUDED)

Symbol	Definition	Units	Fortran Equivalent
Φ_1	Aerodynamic roll angle of fins	radians degrees	CAPHI 1 CP 1
Φ_2	Aerodynamic roll angle of wings	radians degrees	CAPHI 1 CP 2
ψ	Euler angle	degrees	PSI
$\dot{\psi}$	Euler angle rate	rad/sec	PSD

SECTION I

PROGRAM DESCRIPTION

A. INTRODUCTION

The present computer program has resulted from the need for more exact simulation of the motion of dispenser munitions. Particular attention has been given the simulation of the flight of magnus rotor bomblets. The program can also provide trajectory and motion simulation for most unpowered projectiles, missiles, and rockets.

The most significant features of the extended capability trajectory program are:

1. Choice of fixed-plane or body-fixed axes.
2. All-attitude motion prediction.
3. Adaptability to very large spin rates.
4. All basic aeroballistic coefficients are tabular functions of $\tilde{\alpha}$ and M. In addition, spin and magnus coefficients are tabular functions of pd/2V.
5. Inclusion of high order nonlinear damping terms.
6. Aerodynamic dependence upon roll angle is included.
7. Provision for aerodynamic and configurational asymmetries, including c. g. lateral offset.
8. Provision for wind.
9. Provision for elimination of high frequency (nutational) motion.

B. COORDINATE SYSTEMS

Either of two moving axes systems can be selected, a body-fixed coordinate system or a fixed-plane coordinate system. Both sets of moving coordinates have as their origin the center of mass of the body.

The inertial reference system is a set of XYZ non-rotating earth-fixed coordinates, with the origin at sea level. The orientation of the moving axes with respect to the inertial axes is given by three Euler angles: θ , Ψ , and φ , as depicted in Figure 1.

Body-Fixed Coordinates The body-fixed coordinates are comprised of the right-handed orthogonal set of axes, x , y , and z .

Fixed-Plane Coordinates The fixed-plane coordinates are comprised of the right-handed orthogonal set of axes x , y' , and z' . In the fixed-plane system, the y' axis is initially in the XY plane and is so restrained as to stay in that plane. Consequently, $\varphi = \dot{\varphi} = 0$ for the fixed-plane coordinate system.

Since the body can rotate with respect to the x y' z' axes, this axes system is restricted to bodies with rotational symmetry.

C. ATTITUDE REPRESENTATION

The orientation of the moving coordinates, for input and output purposes, is described by the Euler angles. For computational purposes, however, an angular orientation scheme based on quaternions is used. The quaternion system avoids the discontinuities which occur in the trigonometric functions and angular rates when the motion passes from quadrant to quadrant.

The scalar relations between the four parameter quaternion system and the three Euler angles are defined for the body-fixed and fixed-plane system as follows:(1)

Body-fixed axes:

$$\lambda_0 = \cos(\varphi/2)\cos(\theta/2)\cos(\Psi/2) + \sin(\varphi/2)\sin(\theta/2)\sin(\Psi/2)$$

$$\lambda_1 = \sin(\varphi/2)\cos(\theta/2)\cos(\Psi/2) - \cos(\varphi/2)\sin(\theta/2)\sin(\Psi/2)$$

$$\lambda_2 = \cos(\varphi/2)\sin(\theta/2)\cos(\Psi/2) + \sin(\varphi/2)\cos(\theta/2)\sin(\Psi/2)$$

$$\lambda_3 = \cos(\varphi/2)\cos(\theta/2)\sin(\Psi/2) - \sin(\varphi/2)\sin(\theta/2)\cos(\Psi/2)$$

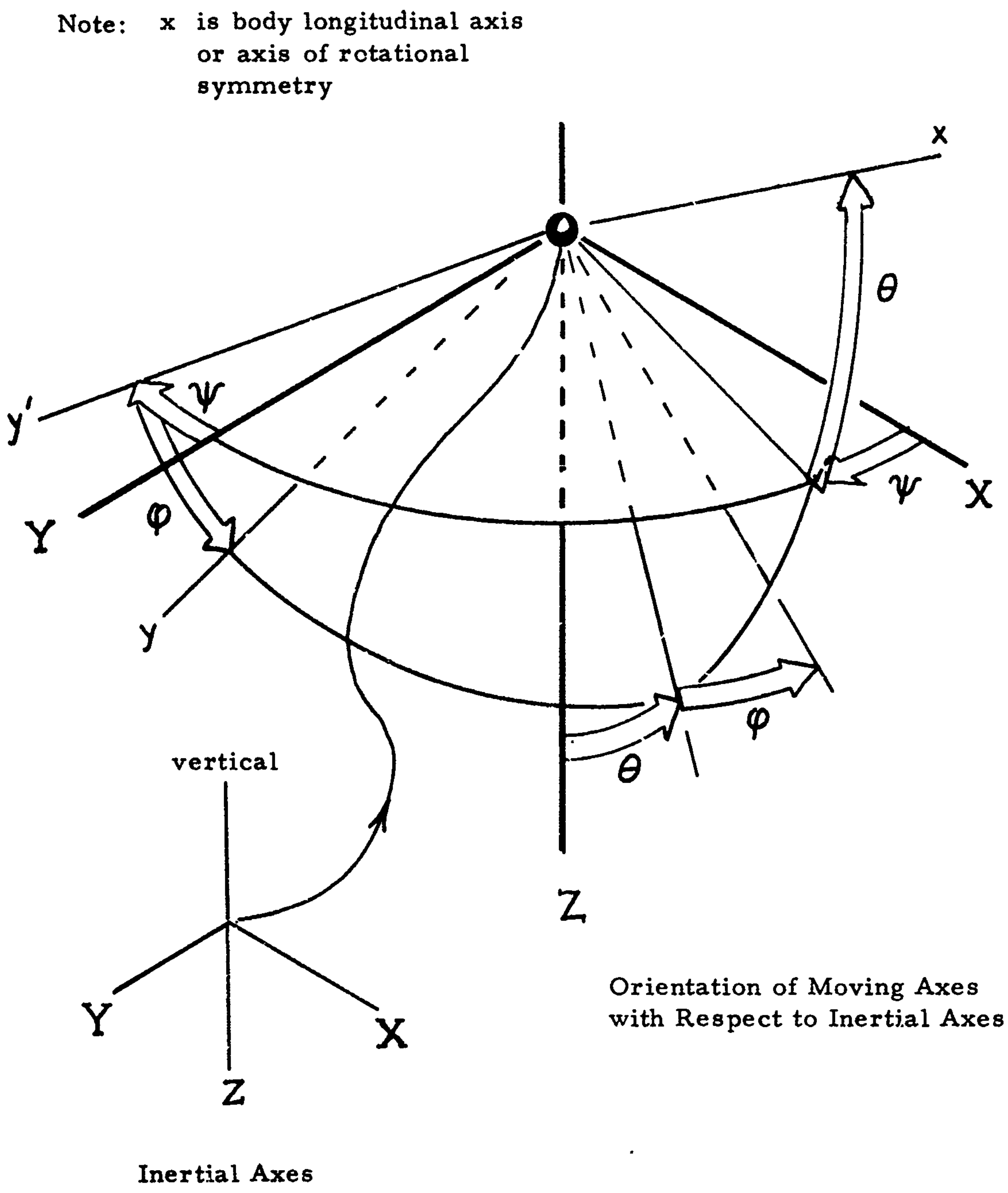


Figure 1. Coordinate Systems

Fixed-plane axes:

$$\lambda_0 = \cos(\theta/2)\cos(\psi/2)$$

$$\lambda_1 = -\sin(\theta/2)\sin(\psi/2)$$

$$\lambda_2 = \sin(\theta/2)\cos(\psi/2)$$

$$\lambda_3 = \cos(\theta/2)\sin(\psi/2)$$

Euler angles:

$$\sin\theta = 2(\lambda_0\lambda_2 - \lambda_1\lambda_3) \qquad ; \ -\pi/2 \le \theta \le \pi/2$$

$$\tan\theta = \frac{2(\lambda_0\lambda_2 - \lambda_1\lambda_3)}{\sqrt{[2(\lambda_2\lambda_3 + \lambda_0\lambda_1)]^2 + [\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2]^2}}$$

$$\tan\psi = \frac{2(\lambda_1\lambda_2 + \lambda_0\lambda_3)}{\lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2} \qquad ; \ -\pi < \psi \le \pi$$

$$\tan\varphi = \frac{2(\lambda_2\lambda_3 + \lambda_0\lambda_1)}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} \qquad ; \ -\pi < \varphi \le \pi$$

<u>Rotation Matrices</u> In the equations of motion, transformations from the moving axes to the fixed inertial axes are required. These can be expressed (for either axis system) by the general quaternion rotation matrix given below, where the subscripts F and M refer to the fixed inertial axes and the moving axis, respectively.

$$\begin{bmatrix} 0 \\ \bar{\xi}_{F_1} \\ \bar{\xi}_{F_2} \\ \bar{\xi}_{F_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \lambda_0^2+\lambda_1^2-\lambda_2^2-\lambda_3^2 & 2(\lambda_1\lambda_2-\lambda_0\lambda_3) & 2(\lambda_1\lambda_3+\lambda_0\lambda_2) \\ 0 & 2(\lambda_1\lambda_2+\lambda_0\lambda_3) & \lambda_0^2-\lambda_1^2+\lambda_2^2-\lambda_3^2 & 2(\lambda_2\lambda_3-\lambda_0\lambda_1) \\ 0 & 2(\lambda_1\lambda_3-\lambda_0\lambda_2) & 2(\lambda_2\lambda_3+\lambda_0\lambda_1) & \lambda_0^2-\lambda_1^2-\lambda_2^2+\lambda_3^2 \end{bmatrix} \begin{bmatrix} 0 \\ \bar{\xi}_{M_1} \\ \bar{\xi}_{M_2} \\ \xi_{M_3} \end{bmatrix}$$

D. BASIC EQUATIONS OF MOTION

The equations of motion consist of 13 differential equations for the variables

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\lambda}_0 \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \dot{\lambda}_3 \end{bmatrix} = \text{Equations of Motion}$$

which must be integrated to obtain the desired motion solution and trajectory. The variables u, v, and w are the linear velocity components in the direction of the moving axes, and p , q , and r are the angular velocity components corresponding to the moving axes. Following standard notation, these components will be with respect to the x , y , z or x , y' , z', depending upon whether the moving axes are body-fixed or fixed-plane, respectively. It follows that p and u are identical in the two coordinate systems, but v , w , q , and r are not.

The derivation of the fixed-plane equations of motion must, of necessity, consider the simplified inertial tensor

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} = \left[I \right]_{F.P.}$$

The body-fixed equations of motion are more general. Provision is made for bodies with an inertia tensor *

$$\begin{bmatrix} I_x & I_{xy} & 0 \\ I_{yx} & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} = \left[\, I \,\right]_{B.F.}$$

corresponding to the xy plane as a plane of mirror symmetry. This is sufficient for most simulation work.

From basic mechanics, we let Ω represent the angular velocity vector of the coordinate system with respect to the inertial system and ω represent the angular velocity of the body with respect to the moving coordinates, and the fundamental equations of motion become (for constant mass and inertia)

$$F = -mg + m\dot{V} + \Omega \times mV$$

$$M = I\dot{\omega} + \Omega \times [I]\omega$$

where $F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{B.F.} = \begin{bmatrix} F_x \\ F_{y'} \\ F_{z'} \end{bmatrix}_{F.P.}$ = aerodynamic force

$$M = \begin{bmatrix} L \\ M \\ N \end{bmatrix}_{B.F.} = \begin{bmatrix} L \\ M \\ M \end{bmatrix}_{F.P.} = \text{aerodynamic moment}$$

$$g = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}_{B.F.} = \begin{bmatrix} g_x \\ g_{y'} \\ g_{z'} \end{bmatrix}_{F.P.} = \text{gravitational acceleration}$$

$$V = \begin{bmatrix} u \\ v \\ w \end{bmatrix}_{B.F.} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}_{F.P.} = \text{velocity}$$

---

* All products of inertia are positive quantities, $I_{xy} = I_{yx}$.

$$\Omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_{B.F.} = \begin{bmatrix} r \tan\theta \\ q \\ r \end{bmatrix}_{F.P.} = \text{angular velocity of coordinates}$$

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_{B.F.} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_{F.P.} = \text{angular velocity of body}$$

and $\tan\theta = \dfrac{2(\lambda_0\lambda_2 - \lambda_1\lambda_3)}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2}$

Substitution of the appropriate quantities into the fundamental equations of motion results in the scalar equations for $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{p}$, $\dot{q}$, and $\dot{r}$. The differential equations for $\dot{X}$, $\dot{Y}$, $\dot{Z}$ are obtained by transformation of the components u, v, w. Finally, the derivatives of the quaternions are obtained from (1)

$$\dot{\lambda} = \frac{1}{2}\lambda * \Omega$$

where * denotes a non-commutative quaternion product,

$$a * b = (a_0 + a_1 i + a_2 j + a_3 k) \quad (b_0 + b_1 i + b_2 j + b_3 k)$$

which can be expressed in matrix form as

$$a * b = \begin{bmatrix} (a_0) & (-a_1) & (-a_2) & (-a_3) \\ (a_1) & (a_0) & (-a_3) & (a_2) \\ (a_2) & (a_3) & (a_0) & (-a_1) \\ (a_3) & (-a_2) & (a_1) & (a_0) \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

The resulting equations of motion are summarized in Tables I and II for the body-fixed and fixed-plane axes, respectively.

Note that a singularity exists in the fixed-plane equations of motion for $\theta = \pm \pi/2$, which precludes the use of these equations of motion when θ may approach $\pi/2$.

TABLE I. BODY-FIXED EQUATIONS OF MOTION

$$\dot u = rv - qw + 2(\lambda_1\lambda_3 - \lambda_0\lambda_2)g + F_x/m$$

$$\dot v = pw - ru + 2(\lambda_2\lambda_3 + \lambda_0\lambda_1)g + F_y/m$$

$$\dot w = qu - pv + (\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2)g + F_z/m$$

$$\dot p = \frac{I_{xy}}{I_x}(\dot q - pr) - \left(\frac{I_z - I_y}{I_x}\right)qr + \frac{L}{I_x}$$

$$= \frac{-[(I_x + I_y - I_z)I_{xy}r]p + [I_{xy}^2 + I_y(I_y - I_z)]q\cdot r + I_y\cdot L + I_{xy}M}{I_xI_y - I_{xy}^2}$$

$$\dot q = \frac{I_{xy}}{I_y}(\dot p + q\cdot r) - \left(\frac{I_x - I_z}{I_y}\right)pr + \frac{M}{I_y}$$

$$= \frac{[(I_x + I_y - I_z)I_{xy}\cdot r]q - [I_{xy}^2 + I_x(I_x - I_z)]p\cdot r + I_{xy}\cdot L + I_x\cdot M}{I_xI_y - I_{xy}^2}$$

$$\dot r = \frac{(p^2 - q^2)I_{xy} + (I_x - I_y)p\cdot q + N}{I_z}$$

$$\dot X = (\lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2)u + 2(\lambda_1\lambda_2 - \lambda_0\lambda_3)v + 2(\lambda_1\lambda_3 + \lambda_0\lambda_2)w$$

$$\dot Y = 2(\lambda_1\lambda_2 + \lambda_0\lambda_3)u + (\lambda_0^2 - \lambda_1^2 + \lambda_2^2 - \lambda_3^2)v + 2(\lambda_2\lambda_3 - \lambda_0\lambda_1)w$$

$$\dot Z = 2(\lambda_1\lambda_3 - \lambda_0\lambda_2)u + 2(\lambda_2\lambda_3 + \lambda_0\lambda_1)v + (\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2)w$$

$$\dot\lambda_0 = \frac{1}{2}(-\lambda_1 p - \lambda_2 q - \lambda_3 r)$$

$$\dot\lambda_1 = \frac{1}{2}(\lambda_0 p - \lambda_3 q + \lambda_2 r)$$

$$\dot\lambda_2 = \frac{1}{2}(\lambda_3 p + \lambda_0 q - \lambda_1 r)$$

$$\dot\lambda_3 = \frac{1}{2}(-\lambda_2 p + \lambda_1 q + \lambda_0 r)$$

TABLE II. FIXED-PLANE EQUATIONS OF MOTION

$$\dot{u} = rv - qw + 2(\lambda_1\lambda_3 - \lambda_0\lambda_2)g + F_x/m$$

$$\dot{v} = 2w\,\frac{\lambda_1\lambda_3 - \lambda_0\lambda_2}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} - r\cdot u + \frac{F_y}{m}$$

$$\dot{w} = qu - 2v\cdot r\,\frac{\lambda_1\lambda_3 - \lambda_0\lambda_2}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} + (\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2)g + \frac{F_3}{m}$$

$$\dot{p} = \frac{L}{I_x}$$

$$\dot{q} = r\left(2\cdot r\,\frac{\lambda_1\lambda_3 - \lambda_0\lambda_2}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} - \frac{I_x}{I}p\right) + \frac{M}{I}$$

$$\dot{r} = -q\left(2\cdot r\,\frac{\lambda_1\lambda_3 - \lambda_0\lambda_2}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} - \frac{I_x}{I}p\right) + \frac{N}{I}$$

$$\dot{X} = (\lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2)u + 2(\lambda_1\lambda_2 - \lambda_0\lambda_3)v + 2(\lambda_1\lambda_3 + \lambda_0\lambda_2)w$$

$$\dot{Y} = 2(\lambda_1\lambda_2 + \lambda_0\lambda_3)u + (\lambda_0^2 - \lambda_1^2 + \lambda_2^2 - \lambda_3^2)v + 2(\lambda_2\lambda_3 - \lambda_0\lambda_1)w$$

$$\dot{Z} = 2(\lambda_1\lambda_3 - \lambda_0\lambda_2)u + 2(\lambda_2\lambda_3 + \lambda_0\lambda_1)v + (\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2)w$$

$$\dot{\lambda}_0 = -\tfrac{1}{2}\left(\lambda_2 q + \frac{\lambda_3 r}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2}\right)$$

$$\dot{\lambda}_1 = -\tfrac{1}{2}\left(\lambda_3 q + \frac{\lambda_2 r}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2}\right)$$

$$\dot{\lambda}_2 = \tfrac{1}{2}\left(\lambda_0 q + \frac{\lambda_1 r}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2}\right)$$

$$\dot{\lambda}_3 = \tfrac{1}{2}\left(\lambda_1 q + \frac{\lambda_0 r}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2}\right)$$

E. BASIC AERODYNAMIC SYSTEM

Basic Aeroballistic Coefficients The aerodynamic forces and moments are expressed in coefficient form, using an aeroballistic system consistent with that which is used for symmetric missiles.(2) The basic aeroballistic coefficients are valid for either the body-fixed or fixed-plane coordinate systems, and only require that the configuration have trigonal or greater rotational symmetry. These basic aeroballistic coefficients are given below:

$C_x \; (\vec{\alpha}, M)$ = axial force coefficient

$C_N \; (\vec{\alpha}, M)$ = normal force coefficient

$C_M \; (\vec{\alpha}, M)$ = overturning moment coefficient

$C_{M_q} (\vec{\alpha}, M)$ = pitch damping coefficient based on $\frac{qd}{2V}$

$C_{N_p} (\vec{\alpha}, M, \frac{pd}{2V}) = -C_{L_p} (\vec{\alpha}, M, \frac{pd}{2V})$ = magnus force coefficient for body-fixed and fixed-plane axes, respectively.

$C_{M_p} (\vec{\alpha}, M, \frac{pd}{2V})$ = magnus moment coefficient

$C_\ell \; (\vec{\alpha}, M, \frac{pd}{2V})$ = spin torque coefficient due to canted fins or vanes

$C_{\ell_p} (\vec{\alpha}, M, \frac{pd}{2V})$ = spin damping coefficient

The above coefficients depend only upon the total angle of attack, $\vec{\alpha}$, and are independent of the components of the angle of attack.* These coefficients are also functions of Mach number, and in addition, C_{N_p}, C_{M_p}, C_ℓ, and C_{ℓ_p} are permitted to be functions of the nondimensional spin parameter, pd/2V.

The sense of the basic aeroballistic forces and moments is indicated in Figure 2. The aeroballistic forces and moments are transformed to the forces and moments corresponding to the moving coordinates

---

* The total angle of attack is the angle between x axis and the aerodynamic velocity vector.

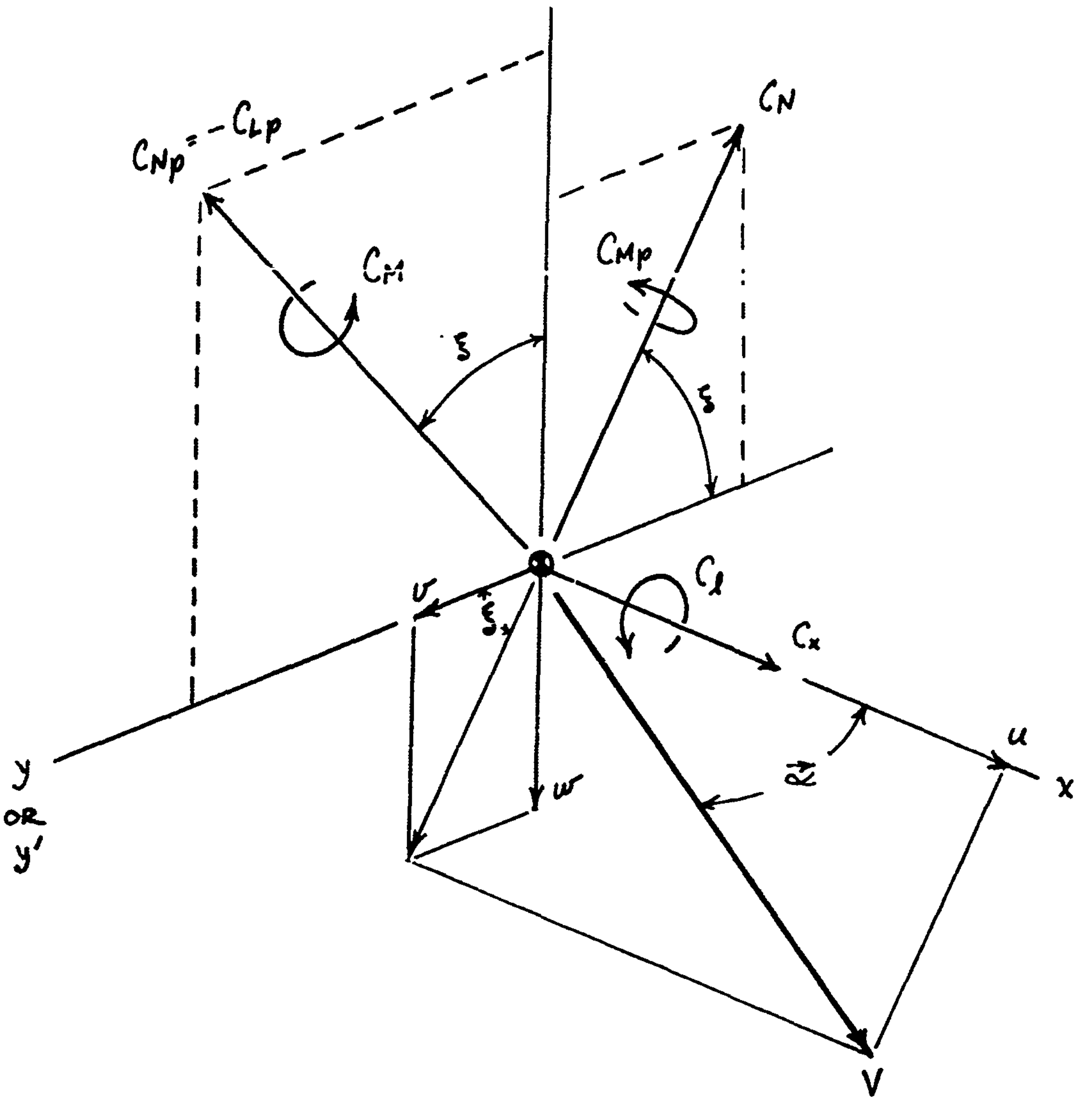


Figure 2. Aeroballistic Force and Moment Definitions

by the matrices

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} F_x \\ F_{y'} \\ F_{z'} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & (-\cos\xi) & (\sin\xi) \\ 0 & (-\sin\xi) & (-\cos\xi) \end{bmatrix} \begin{bmatrix} C_x \\ C_N \\ C_{N_p} \end{bmatrix} \cdot \frac{1}{2}\rho V^2 S$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & (\sin\xi) & (\cos\xi) \\ 0 & (-\cos\xi) & (\sin\xi) \end{bmatrix} \begin{bmatrix} C_\ell \\ C_M \\ C_{M_p} \end{bmatrix} \cdot \frac{1}{2}\rho V^2 Sd$$

where

$$\cos\xi = \frac{v}{\sqrt{v^2 + w^2}}$$

$$\sin\xi = \frac{w}{\sqrt{v^2 + w^2}}$$

<u>Additional Damping Parameters</u> In general, the cross-angular velocity does not coincide with the plane of the total angle of attack. For these nonplanar motions, it is now generally accepted that the aerodynamic damping can differ from that for a planar motion.[3] The present trajectory program accounts for the nonplanar motion damping by dividing the cross-angular velocity into components q' and r', which are coincident with, and normal to, the angle of attack plane, respectively. The nonplanar damping contribution due to r' is incorporated by use of an additional damping coefficient, C_{n_r}. In a similar manner, the coupling coefficients, $C_{m_{p_r}}$ and $C_{n_{pq}}$, are introduced. The sense of these coefficients is depicted in Figure 3.

The moments corresponding to the moving axes are determined by the double rotation matrices:

$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} (\sin\xi) & (\cos\xi) \\ (-\cos\xi) & (\sin\xi) \end{bmatrix} \begin{bmatrix} (\sin\xi) & (-\cos\xi) \\ (\cos\xi) & (\sin\xi) \end{bmatrix} \begin{bmatrix} (\frac{qd}{2V}) \cdot C_{m_q} \\ (\frac{rd}{2V}) \cdot C_{n_r} \end{bmatrix} \cdot \frac{1}{2}\rho V^2 Sd$$

$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{vmatrix} (\sin\xi) & (\cos\xi) \\ (-\cos\xi) & (\sin\xi) \end{vmatrix} \begin{vmatrix} (\sin\xi) & (\cos\xi) \\ (-\cos\xi) & (\sin\xi) \end{vmatrix} \begin{vmatrix} (\frac{rd}{2V}) \cdot C_{m_{p_r}} \\ (\frac{qd}{2V}) \cdot C_{n_{pq}} \end{vmatrix} \cdot \frac{1}{2}\rho V^2 Sd$$

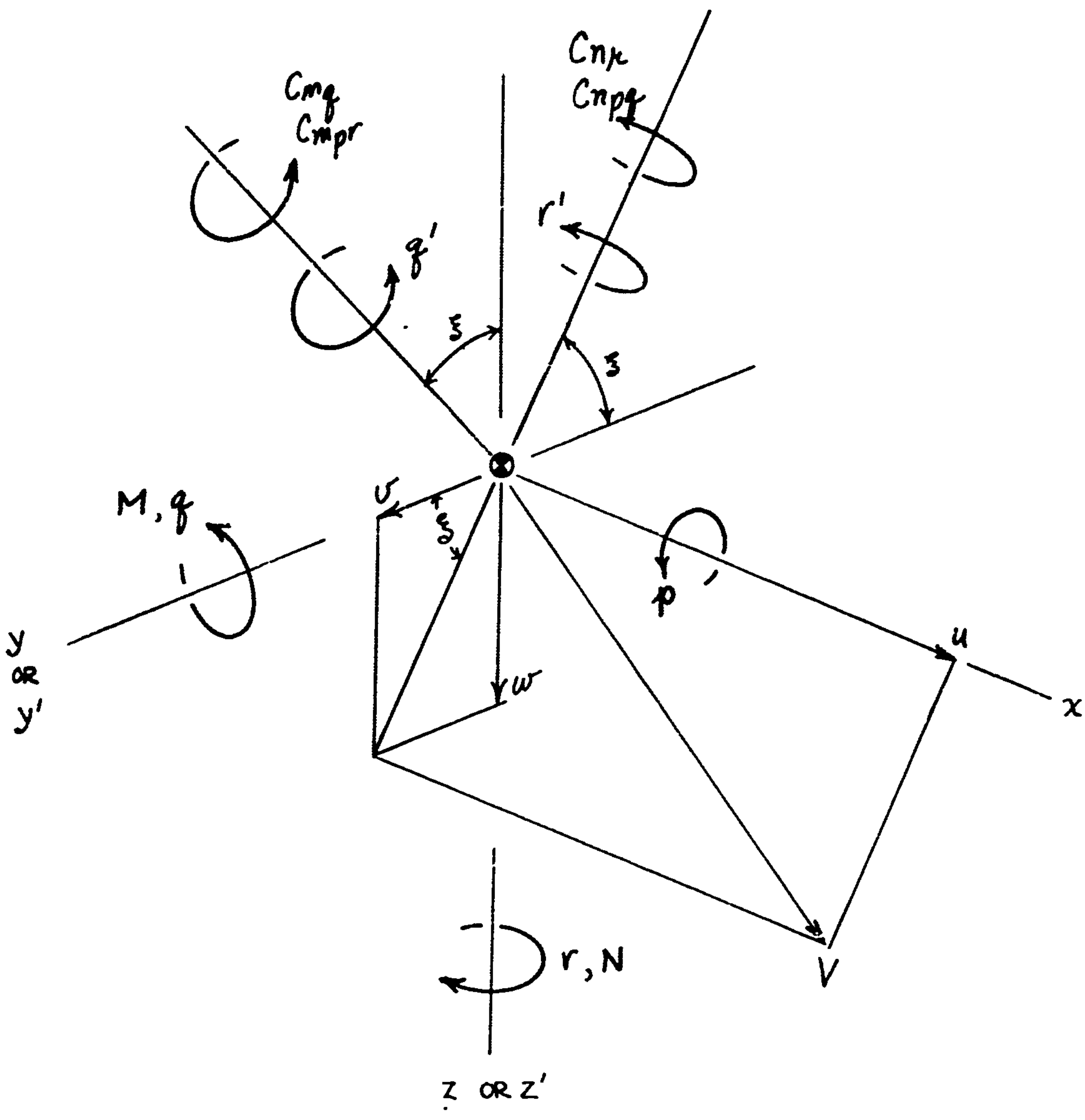


Figure 3. Damping Parameter Definitions

The damping coefficients, C_{n_r}, $C_{m_{p_r}}$, and $C_{n_{p_q}}$, are functions of total angle of attack and Mach number, as summarized below:

$C_{n_r}\ (\vec{\alpha}, M)$ = non planar damping coefficient

$C_{m_{p_r}}\ (\vec{\alpha}, M)$ = cross damping moment - angle of attack plane

$C_{n_{p_q}}\ (\vec{\alpha}, M)$ = cross damping moment - magnus plane.

These coefficients, together with the basic aeroballistic coefficients, comprise the aerodynamic coefficient system for the fixed-plane-axes system.

F. EXTENDED AERODYNAMIC SYSTEM

When the body-fixed-axes option is selected, additional aerodynamic coefficients are included to account for body-fixed asymmetries, windward meridian orientation (aerodynamic roll angle), flow asymmetry, and lateral displacement of the center of gravity. These coefficients are depicted schematically in Figure 4.

Body-Fixed Asymmetries Body-fixed aerodynamic forces and moments due to misalignment, cant, or asymmetry of body and/or lifting surfaces are accounted for by the coefficients

$C_{y_o}\ (\vec{\alpha}, M)$

$C_{z_o}\ (\vec{\alpha}, M)$

$C_{m_o}\ (\vec{\alpha}, M)$

$C_{n_o}\ (\vec{\alpha}, M)$.

The aerodynamic effects of misalignment, cant, and asymmetry on the axial force and rolling moment are accounted for by the basic aeroballistic coefficients $C_x\ (\vec{\alpha}, M)$ and $C_\ell\ (\vec{\alpha}, M)$.

Aerodynamic Effects Due to Windward Meridian Orientation. Bodies with wings and fins are subject to aerodynamic effects related to the orientation of the aerodynamic surfaces with respect to the windward meridian of the cross flow. The orientation of the aerodynamic surfaces with respect to the cross flow is specified by the aerodynamic roll angle Φ. This angle is defined as a clockwise rotation of the aerodynamic surfaces with respect to the cross flow, looking in the direction of the x axis, as described by the following sketch. The orientation of the symmetry planes

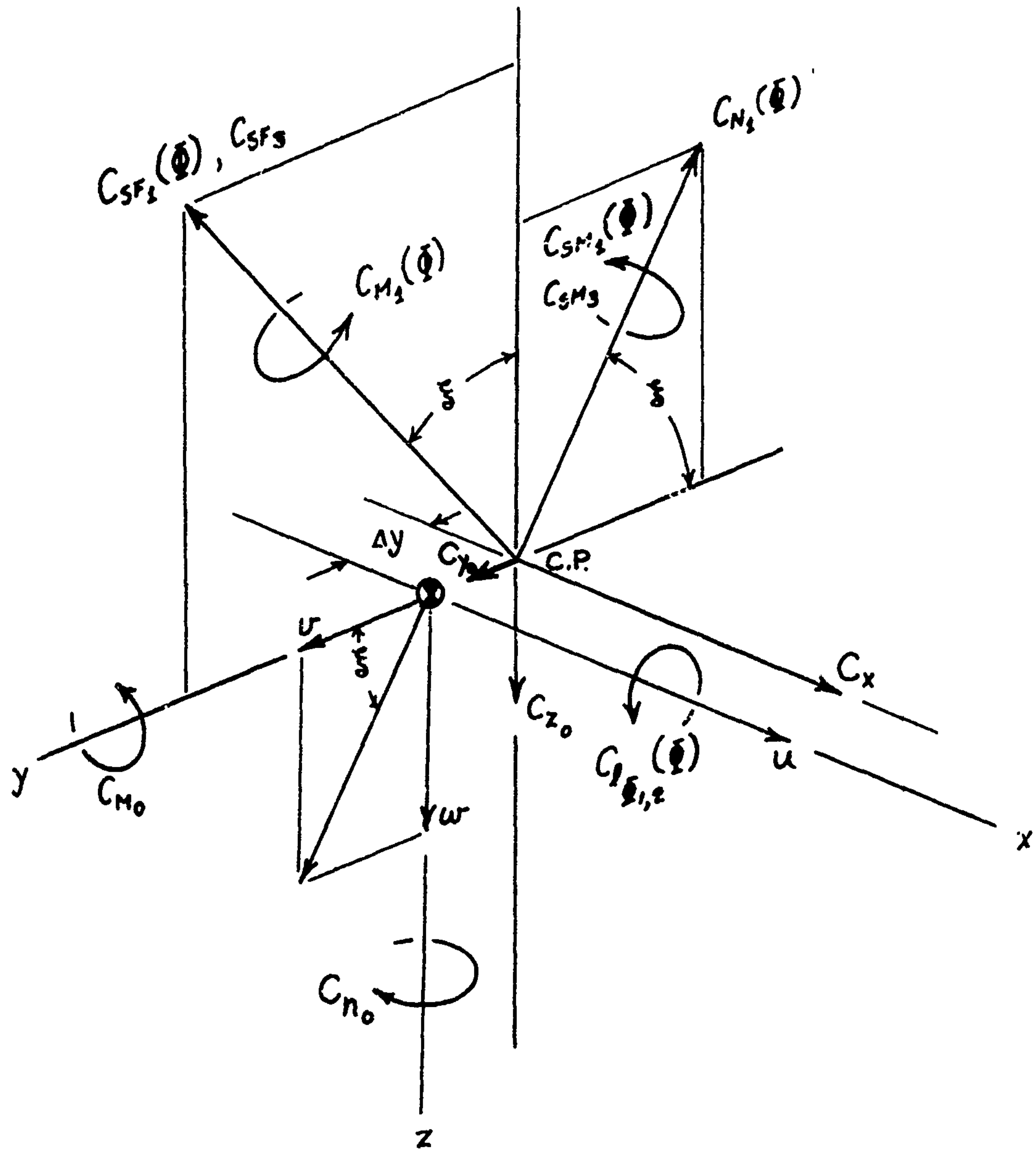
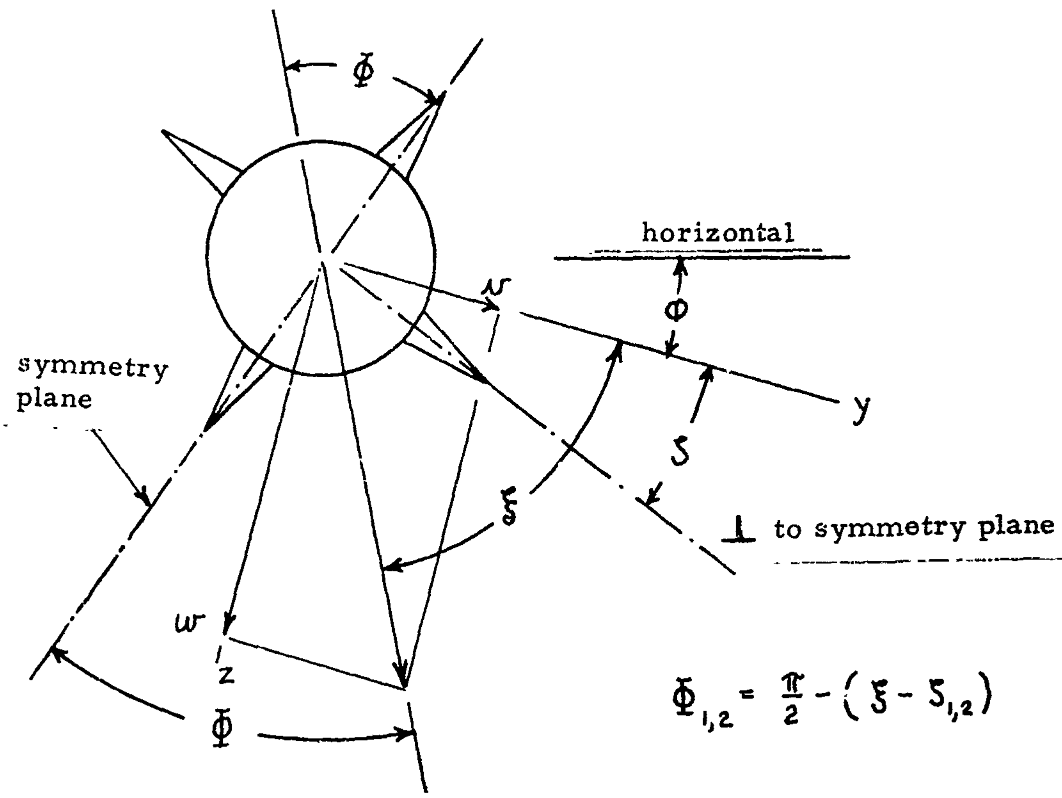


Figure 4. Additional Aerodynamic Forces and Moments for the Body-Fixed Axes Option

of the fins or wings with respect to arbitrary body axes is given by the phase angle ζ . This phase angle may be necessary if body-fixed asymmetries, such as lateral c.g. offset, are also present.



The most significant induced aerodynamic effects resulting from windward meridian are generally the induced rolling moment, the induced side force, and the induced side moment. The two moments often have a profound effect on vehicle stability, and are usually of greatest magnitude for 3 or 4-fin tail assemblies. Likewise, very large induced aerodynamic effects would be expected from a planar surface, such as a wing. In addition, the aerodynamic roll angle will have some effect on the normal force and pitching moment, which must be considered for completeness.

For a single set of aerodynamic surfaces of uniform size, the functional dependence of the aerodynamic forces and moments on the aerodynamic roll angle is given by the following coefficients and functional relationships:

$$C_{SF_1}(\vec{\alpha}, M, \Phi) = C_{SF_1}(\vec{\alpha}, M) \sin(\eta_1, \Phi_1)$$

$$C_{N_1}(\vec{\alpha}, M, \Phi) = C_{N_1}(\vec{\alpha}, M) \sin(\eta_1, \Phi_1)$$

$$C_{SM_1}(\vec{\alpha}, M, \Phi) = C_{SM_1}(\vec{\alpha}, M) \sin(\eta_1, \Phi_1)$$

$$C_{M_1}(\vec{\alpha}, M, \Phi) = C_{M_1}(\vec{\alpha}, M) \sin(\eta_1, \Phi_1)$$

$$C_{l_{\Phi_1}}(\vec{\alpha}, M, \Phi) = C_{l_{\Phi_1}}(\vec{\alpha}, M) \sin(\eta_1, \Phi_1)$$

where η = number of axially symmetric fins (i.e., cruciform fins, $\eta = 4$).

Although the dependence of C_{SF_1}, C_{N_1}, C_{SM_1}, C_{M_1}, and $C_{l_{\Phi_1}}$ on Φ could be more general, only the first harmonic is used. The principal reasons for this simplification is the usual lack of more definitive wind tunnel data (i.e., usual practice is to measure the near maximum effect of Φ by testing at $\Phi = \pi/2\eta$).

Limited provision is made for a second set of aerodynamic surfaces with a different η. For the second set of aerodynamic surfaces only the coefficient $C_{l_{\Phi_2}}$ is provided.

Experience has shown that, in general, one set of aerodynamic surfaces will have aerodynamic induced effects which are much larger, and the complete force and moment coefficients should be used with this set of aerodynamic surfaces.

Combined Effect of Geometric Asymmetry and Windward Meridian Orientation The preceding paragraphs describe how provision has been made for both the body-fixed aerodynamic coefficients C_{y_o}, C_{z_o}, C_{m_o}, and C_{n_o} and windward meridian orientation. In practice, a single vehicle modification (or asymmetry) may lead to both effects simultaneously. And it is important to recognize, clearly, how the aerodynamic coefficients should be interpreted.

Consider a body with a portion of the nose sliced off as in Figure 5, such that the xz plane is a plane of mirror symmetry. Then, in addition to the body-fixed trim force $C_{z_o}(\vec{\alpha})$ and the normal force $C_N(\vec{\alpha})$ it is possible that induced aerodynamic normal and side forces exist. These are $C_{N_1}(\vec{\alpha}, \Phi)$ and $C_{SF_1}(\vec{\alpha}, \Phi)$, respectively, and have the vector orientations indicated by Figure 5. Note further that in Figure 5, which depicts a case for which $\eta = 1$, the first harmonic relationship used for C_{N_1} and C_{SF_1} requires that $C_{N_1} = C_{SF_1} \equiv 0$ at $\Phi = 0, \pi$.

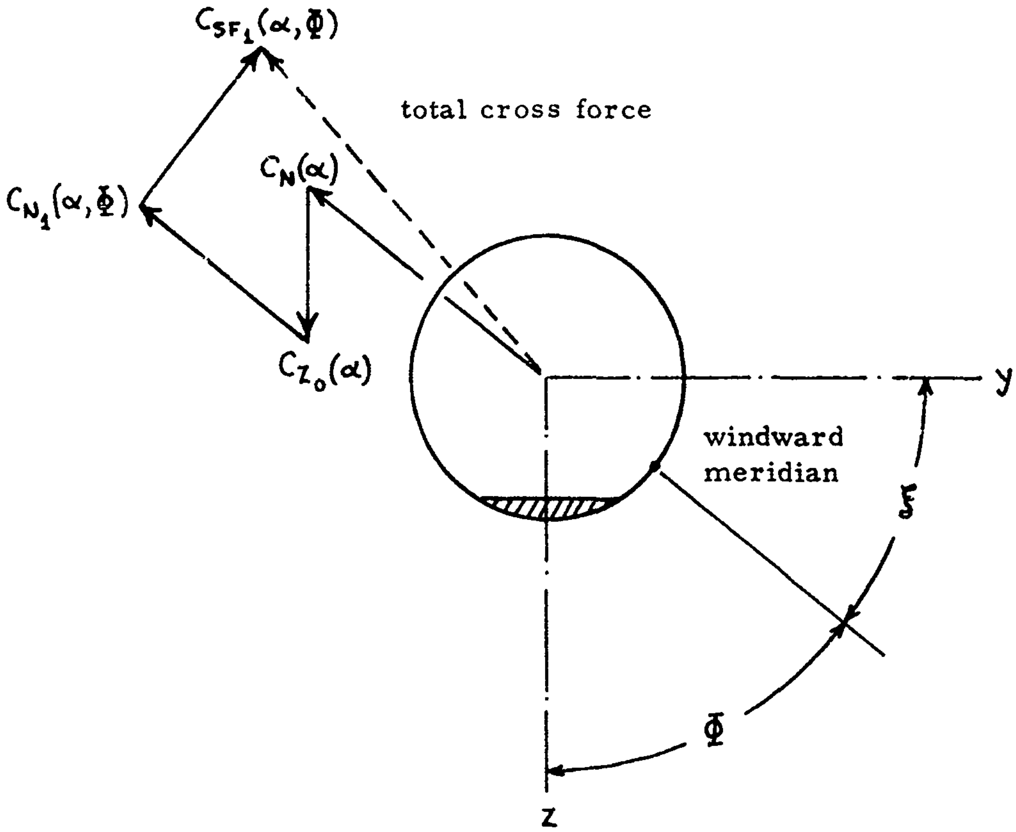


Figure 5. Aerodynamic Cross Force Due to Combined Effect of Geometric Asymmetry and Windward Meridian Orientation

At this point, it is very important to observe that the effect of the body-fixed trim force may be incorrectly interpreted as having a Φ dependence. For example, if the y z axes are fixed and there is a rotation of the windward meridian, then there ill occur a corresponding angular rotation between the vectors $C_N(\tilde{\alpha})$ and $C_{z_o}(\tilde{\alpha})$, such that the magnitude of the total cross force will vary with Φ. This effect is only due to the variation of the normal force with Φ, and is correctly accounted for in the body-fixed equations of motion, due to the fact that $F_y \propto = C_N(\tilde{\alpha}) \cos \xi$ and $F_z \propto = C_N(\tilde{\alpha}) \sin \xi$. Thus, in reducing wind tunnel force measurements to determine the effect of Φ, when configurational asymmetries are present, the trim force coefficients C_{y_o} and C_{z_o} must be subtracted first. Similar arguments apply to the aerodynamic moment coefficients.

Additional Aeroballistic Coefficients for Flow Asymmetry
The aerodynamic forces on a pure axi-symmetric body (i.e., body of revolution) with zero spin, are not always symmetric with respect to the angle of attack plane as postulated in the basic aeroballistic formulation. In particular, it has been noted that long pointed-nose bodies tend to have an asymmetric vortex-wake structure at large angles of attack for a range of Reynolds numbers and Mach numbers. Once established, the asymmetrical wake becomes reasonably stable, such that the resulting forces and moments can be considered steady. To include the above effects in the aerodynamic model, the following coefficients and functional relationships are added to the aeroballistic system employed with the body-fixed axes:

$$C_{SF_3}(\tilde{\alpha}, M)$$
$$C_{SM_3}(\tilde{\alpha}, M)$$

It will be noted that these coefficients have the same vector orientation as the magnus force and moment.

Effect of Lateral Displacement of Center of Gravity from the Longitudinal Reference Axis Additional aerodynamic moment terms are introduced into the equations of motion if the center of gravity is laterally displaced from the longitudinal reference axis.

These moments are:

$$\Delta C_\ell = -\left(\sum C_z\right)(\Delta y/d)$$

$$\Delta C_n = C_x(\tilde{\alpha}, M)(\Delta y/d)$$

Only a c.g. offset along the y body-fixed axis will be considered, because all other body-fixed forces and moments are introduced with complete generality. Also, the c.g. offset conforms to the xy plane, which is assumed to be a plane of mirror symmetry.

G. SUMMARY OF AERODYNAMIC FORCES AND MOMENTS

Table III summarizes the aerodynamic forces and moments, in scalar form, for each degree of freedom. Those terms which are added for the body-fixed axes option are enclosed by dashed lines.

H. ATMOSPHERIC MODEL AND WIND SIMULATION

Atmospheric Model   Air density, ρ , and velocity of sound, a , are approximated for standard day conditions by the relations:

$$\rho = 0.0023769\,[1 + 6.875 \times 10^{-6}(Z)]^{4.2561} \qquad (-Z) < 36,000 \text{ ft.}$$

$$\rho = 0.0040\, e^{4.806 \times 10^{-5}(Z)} \qquad (-Z) > 36,000 \text{ ft.}$$

$$a = 968.46 - 0.004123(-Z) \qquad (-Z) < 36,000 \text{ ft.}$$

$$a = 968.46 \qquad (-Z) > 36,000 \text{ ft.}$$

I. ATMOSPHERIC WIND SIMULATION

Atmospheric wind simulation is accomplished by introducing wind vectors $\dot{X}_W$, $\dot{Y}_W$, as a function of altitude ($-Z$). The aerodynamic velocity components of the body with respect to the moving air mass are

$$(\dot{X} - \dot{X}_W),\ (\dot{Y} - \dot{Y}_W),\ \dot{Z}$$

and the corresponding aerodynamic velocity components along the body-axes are designated u_A , v_A , w_A . In the presence of wind, the aerodynamic forces and moments are redefined so as to be functions of

$$V_A = \sqrt{u_A^2 + v_A^2 + w_A^2}$$

$$\alpha = \cos^{-1} \frac{u_A}{V_A}$$

$$M = \frac{V_A}{a}$$

$$\xi = \tan^{-1} \frac{w_A}{v_A}$$

TABLE III. AERODYNAMIC FORCE AND MOMENT EXPANSIONS

$$F_x = \frac{1}{2}\rho V^2 S\Big\{ C_X(\tilde{\alpha}, M)\Big\}$$; Body-Fixed or Fixed Plane Axes

$\boxed{}$ Denotes Body-Fixed Axes Only

$$F_y = F_{y'} = \frac{1}{2}\rho V^2 S\Big\{\Big[-C_N(\tilde{\alpha},M) - \boxed{C_{N_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1]}\Big]\cos\xi + \Big\{\Big[\frac{-C_{L_p}}{\boxed{C_{N_p}}}\big(\tilde{\alpha},M,\tfrac{pd}{2V}\big)\Big]\Big[\frac{pd}{2V}\Big] + \boxed{C_{SF_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1] + C_{SF_3}(\tilde{\alpha},M)}\Big\}\sin\xi + \boxed{C_{y_0}(\tilde{\alpha},M)}\Big\}$$

$$F_z = F_{z'} = \frac{1}{2}\rho V^2 S\Big\{\Big[-C_N(\tilde{\alpha},M) - \boxed{C_{N_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1]}\Big]\sin\xi - \Big\{\Big[\frac{-C_{L_p}}{\boxed{C_{N_p}}}\big(\tilde{\alpha},M,\tfrac{pd}{2V}\big)\Big]\Big[\frac{pd}{2V}\Big] + \boxed{C_{SF_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1] + C_{SF_3}(\tilde{\alpha},M)}\Big\}\cos\xi + \boxed{C_{z_0}(\tilde{\alpha},M)}\Big\}$$

$$\begin{aligned} L = \frac{1}{2}\rho V^2 S d\Big\{ & C_\ell\big(\tilde{\alpha},M,\tfrac{pd}{2V}\big) + \Big[C_{\ell_p}\big(\tilde{\alpha},M,\tfrac{pd}{2V}\big)\Big]\Big[\frac{pd}{2V}\Big] + C_{\ell_{\Phi_1}}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1] + C_{\ell_{\Phi_2}}(\tilde{\alpha},M)\sin[\eta_2 \Phi_2] \\ & + \frac{\Delta y}{d}\Big\{\Big[C_N(\tilde{\alpha},M) + C_{N_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1]\Big]\sin\xi + \Big[\Big[C_{N_p}\big(\tilde{\alpha},M,\tfrac{pd}{2V}\big)\Big]\Big[\frac{pd}{2V}\Big] + C_{SF_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1] \\ & \qquad + C_{SF_3}(\tilde{\alpha},M)\Big]\cos\xi - C_{z_0}\Big\}\Big\} \end{aligned}$$

$$\begin{aligned} M = \frac{1}{2}\rho V^2 S d\Big\{ & \Big[C_M(\tilde{\alpha},M) + \boxed{C_{M_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1]}\Big]\sin\xi + \Big[C_{M_p}\big(\tilde{\alpha},M,\tfrac{pd}{2V}\big)\Big]\Big[\frac{pd}{2V}\Big]\cos\xi + \Big[C_{m_q}(\tilde{\alpha},M)\Big]\Big[\Big[\frac{qd}{2V}\Big]\sin^2\xi - \Big[\frac{rd}{2V}\Big]\cos\xi\sin\xi\Big] \\ & + \Big[C_{n_r}(\tilde{\alpha},M)\Big]\Big[\Big[\frac{qd}{2V}\Big]\cos^2\xi + \Big[\frac{rd}{2V}\Big]\sin\xi\cos\xi\Big] + \Big[C_{m_{pr}}(\tilde{\alpha},M)\Big]\Big[\frac{pd}{2V}\Big]\Big[\Big[\frac{qd}{2V}\Big]\cos\xi\sin\xi + \Big[\frac{rd}{2V}\Big]\sin^2\xi\Big] + \Big[C_{n_{pq}}(\tilde{\alpha},M)\Big]\Big[\frac{pd}{2V}\Big]\Big[\Big[\frac{qd}{2V}\Big]\sin\xi\cos\xi - \Big[\frac{rd}{2V}\Big]\cos^2\xi\Big] \\ & + \boxed{\Big[C_{SM_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1] + C_{SM_3}(\tilde{\alpha},M)\Big]\cos\xi + C_{M_0}(\tilde{\alpha},M)}\Big\} \end{aligned}$$

$$\begin{aligned} N = \frac{1}{2}\rho V^2 S d\Big\{ & \Big[-C_M(\tilde{\alpha},M) - \boxed{C_{M_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1]}\Big]\cos\xi + \Big[C_{M_p}\big(\tilde{\alpha},M,\tfrac{pd}{2V}\big)\Big]\Big[\frac{pd}{2V}\Big]\sin\xi - \Big[C_{m_q}(\tilde{\alpha},M)\Big]\Big[\Big[\frac{qd}{2V}\Big]\sin\xi\cos\xi - \Big[\frac{rd}{2V}\Big]\cos^2\xi\Big] \\ & + \Big[C_{n_r}(\tilde{\alpha},M)\Big]\Big[\Big[\frac{qd}{2V}\Big]\cos\xi\sin\xi + \Big[\frac{rd}{2V}\Big]\sin^2\xi\Big] - \Big[C_{m_{pr}}(\tilde{\alpha},M)\Big]\Big[\frac{pd}{2V}\Big]\Big[\Big[\frac{qd}{2V}\Big]\cos^2\xi + \Big[\frac{rd}{2V}\Big]\sin\xi\cos\xi\Big] + \Big[C_{n_{pq}}(\tilde{\alpha},M)\Big]\Big[\frac{pd}{2V}\Big]\Big[\Big[\frac{qd}{2V}\Big]\sin^2\xi - \Big[\frac{rd}{2V}\Big]\cos\xi\sin\xi\Big] \\ & + \boxed{\Big[C_{SM_1}(\tilde{\alpha},M)\sin[\eta_1 \Phi_1] + C_{SM_3}(\tilde{\alpha},M)\Big]\sin\xi + C_{n_0}(\tilde{\alpha},M) + C_z(\tilde{\alpha},M)\Big[\frac{\Delta y}{d}\Big]}\Big\} \end{aligned}$$

where

$$\sin \xi = \frac{w_A}{\sqrt{v_A^2 + w_A^2}}$$

$$\cos \xi = \frac{v_A}{\sqrt{v_A^2 + w_A^2}}$$

The above relations are of the same algebraic form as in the case of zero wind.

The transformation from the inertial wind aerodynamic velocities to the body-axes aerodynamic velocities is given by

$$\begin{bmatrix} u_A \\ v_A \\ w_A \end{bmatrix} = \begin{bmatrix} \lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2 & 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & \lambda_0^2 - \lambda_1^2 + \lambda_2^2 - \lambda_3^2 & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) & \lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2 \end{bmatrix} \begin{bmatrix} \dot{X} - \dot{X}_W \\ \dot{Y} - \dot{Y}_W \\ \dot{Z} - \dot{Z}_W \end{bmatrix}$$

J. QUATERNION NORMALIZATION AND QUATERNION ERROR

The separate integration of each quaternion element, $\lambda_0 \rightarrow \lambda_3$, leads to the possibility of small errors, such that $\lambda_0^2 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \neq 1$. To insure normalization $(\lambda_0^2 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \equiv 1)$, the following correction is made to each quaternion element after integration:

$$\lambda_j^* = \frac{\lambda_j}{\sqrt{\lambda_0^2 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

The above correction scheme is derived from the transformation relationships between the quaternions and the three Euler angles, which uniquely describe the moving-axis-system orientation. The method results in an exact normalized quaternion. However, in the case of the fixed-plane axes, the additional identity, $\lambda_0\lambda_1 + \lambda_2\lambda_3 \equiv 0$, which applies for this case, may not be satisfied. Consequently, the body-fixed axes should be used in preference to the fixed-plane axes whenever the rotational rates and frequencies permit.

The quaternion error, ϵ , is defined as the vector sum of the errors of the normalized quaternion elements after successive predict-correct or correct and re-correct integration operations: i.e.,

$$\epsilon = \sqrt{\sum \left[(\lambda_i)_{SAVE} - (\lambda_i)_o \right]^2}$$

where

$(\lambda_i)_{SAVE}$ = normalized value of λ_i after prediction (or correction)

$(\lambda_i)_o$ = normalized value of λ_i after correction (or re-correction)

K. INTEGRATION

Since the differential equations are quite complicated and lengthy, a refined integration scheme is employed. The basic system utilizes Milne's four-point method of prediction and Simpson's rule for correction. The Runge-Kutta method of third order accuracy is used to calculate the second through fourth ordinates needed to start Milne's method. These are described in detail in the flow charts, Section IV.

Options are also provided for the Adams and trapezoidal integration schemes, using four and three ordinates, respectively.

As a further means of reducing integration error, an additional option is provided for up to n corrections and re-corrections, depending upon the magnitude of the quaternion error, ϵ , as defined in the previous section. If $\epsilon > \epsilon_{max}$, additional corrections and re-corrections will be made until the number of corrections (NCOR) equals the specified maximum number of corrections (NCMAX).

The accumulative integration error for selected variables is provided in the optional program output, based on the errors computed in the last correction or re-correction.

L. OPTIONAL APPROXIMATE EQUATIONS OF MOTION FOR USE WHEN THE NUTATION IS DAMPED

Approximate equations of motion, with $\dot{q} = \dot{r} = 0$, may be selected at a particular time, if the fixed-plane axes option is exercised. In effect, this operation eliminates the high frequency oscillatory motion about the lateral axes, which is usually associated with the nutational mode.

The angular velocities q and r are determined by solution of the differential equations which result with $\dot{q} = \dot{r} = 0$. These modified equations are:

$$r = \frac{\frac{I_x}{I} p - \left[\left(\frac{I_x}{I} p \right)^2 - 8 \left(\frac{\lambda_1 \lambda_3 - \lambda_0 \lambda_2}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} \right) \frac{M}{I} \right]^{1/2}}{4 \left(\frac{\lambda_1 \lambda_3 - \lambda_0 \lambda_2}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} \right)}$$

$$q = - \frac{N \cdot r}{M}$$

The minus sign is selected for the radical, corresponding to the slow precessional mode. The positive root represents a flat spin mode with large angular rates and is of little interest.

The optional equations of motion for $\dot{q} = \dot{r} = 0$ are initiated at a time when the nutational oscillations are known to be damped to an amplitude of a few degrees. The resulting motion predictions are valid to the extent that the nutational motion can be neglected. The approximations are invalid if the high frequency motion is undamped.

Because the angular rates will usually be small with $\dot{q} = \dot{r} = 0$, a larger integration interval can be utilized, and an option for this new time interval is included in the program input.

M. AUXILIARY FUNCTIONS

For purposes of program output, the Euler angle rates are specified, although these quantities do not enter into the equations of motion. The following relationships are utilized for computing the Euler angle time derivatives:

$$\dot{\psi} = \frac{2(\lambda_2 \lambda_3 + \lambda_0 \lambda_1) \cdot q + (\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2) \cdot r}{1 - \left[2(\lambda_1 \lambda_3 - \lambda_0 \lambda_2) \right]^2}$$

$$\dot{\theta} = \frac{(\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2)\cdot q \; - 2(\lambda_2\lambda_3 + \lambda_0\lambda_1)\cdot r}{\left[\, 1 - \left[2(\lambda_1\lambda_3 - \lambda_0\lambda_2)\right]^2 \,\right]^{1/2}}$$

$$\dot{\varphi} = p - 2(\lambda_1\lambda_3 - \lambda_0\lambda_2)\cdot\dot{\psi} \qquad \text{body-fixed axes}$$

$$\dot{\varphi} = \frac{2(\lambda_1\lambda_3 - \lambda_0\lambda_2)\cdot r}{\lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2} - 2(\lambda_1\lambda_3 - \lambda_0\lambda_3)\cdot\dot{\psi} \qquad \text{fixed-plane axes}$$

(The reverse of this page is blank)

SECTION II

INPUT FORMAT AND USER'S INSTRUCTIONS

PHYSICAL MAGNETIC TAPE A2

SYSTEM INPUT (A2)

LEAD CARD SETUP

```
 1. LEAD CARD 1 (ALWAYS)- RUN DESCRIPT ,AXIS SYST AND PRINT OPTIONS
 2. SUBSCRIPT CARD AND INTEGRATION OPTIONS
 3. ALPHA VALUES
 4. MACH  VALUES
 5. PD/2V VALUES
 6. BASIC AEROBALLISTIC COEFFICIENT TABLES
       CX,CN,CM,CMQ,CNR,CMPP,CNPQ,CNP,CL ,CLP,CMP
 7. LEAD CARD 2
       G,DMM,DIX,DI,DIZ,DIXY
 8. LEAD CARD 3
       S,DEE,THETA,PSI,Z,PHI
 9. LEAD CARD 4
       XDOT,YDOT,ZDOT,P,Q,R
10. LEAD CARD 5
       TSTEP,INCPT,TMAX,ZSTOP,TCH,TNEW
11. WIND CARDS     (IF WIND OPTION SELECTED ON SUBSCRIPT CARD)
12. LEAD CARD 6    (IF BODY AXES)
13. ADDITIONAL AERODYNAMIC COEFFICIENT TABLES (IF BODY AXES)
       CYO,CZO,CMO,CNO,CSF1,CN1,CSF3,CSM1,CM1,CSM3,CLPH1,CLPH2
14. LEAD CARD 7  (OMIT IF NO CASE FOLLOWS)
```

SUBSCRIPT CARD AND INTEGRATION OPTIONS
(OMIT IF COLUMN 1 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM			SCALING
1-2	IMAX	=	NUMBER OF ANGLES OF ATTACK, (MAX OF 20)	I2
3-4	JMAX	=	NUMBER OF MACH VALUES (MAX OF 5)	I2
5-6	KMAX	=	NUMBER OF SPIN (PD/2V)VALUES (MAX OF 5)	I2
7-8	LMAX	=	NUMBER OF WIND VALUES (MAX OF 10)	I2
9-10	NRK	=	OPTION FOR INTEGRATION ROUTINE 2=TRAPEZOID ,3=ADAMS,4 OR 0 =MILNES	I2
11-12	NCMAX	=	MAXIMUM NUMBER OF CORRECTIONS	I2
13-24	EPSMAX	=	MAXIMUM ERROR ON ABSOLUTE VALUE OF GENERAL NORMALIZED QUATERNION	F12.5

NOTE- IF IMAX,JMAX,KMAX,LMAX EQUALS ONE,NO INTERPOLATION DONE IN THAT DIMENSION ,HOWEVER, IF IMAX,ETC = 1, CORRESPONDING ANGLE OF ATTACK VALUE CARD,ETC MUST BE INCLUDED WITH ZERO VALUES

NOTE-EPSMAX = 0.00001 WILL BE SATISFACTORY FOR MOST CASES

NOTE- IF LMAX IS ZERO , DELETE WIND TABLES

LEAD CARD 1

COLUMN		ITEM	SCALING
1-3	IRUN	= RUN NUMBER (IDENTIFICATION)	I3
4-5	BLANK		
6-71	HEADER	= ANY LEGAL ALPHA-NUMERICAL DATA.	11A6
72	BLANK		
73	KA6	= OPTION FOR INTERMEDIATE O/P ON A6. (OPTIONAL PRINT OUT)	I1
		= 0 NO INTERMEDIATE O/P ON A6.	
		= 1 INTERMEDIATE O/P ON A6.	
74	BODFIX	= OPTION FOR AXIS SYSTEM (T FOR BODY FIXED, F OR BLANK FOR FIXED PLANE)	L1
75-78	IDATE	= DATE (MONTH, DAY)	I4

IRUN MUST BE RIGHT ADJUSTED.

NOTE THAT BASIC AEROBALLISTIC MAGNUS FORCE COEFFICIENT IS NOT THE SAME FOR BODY-FIXED AND FIXED-PLANE AXES. ALL OTHER COEFFICIENTS CAN BE USED INTERCHANGEABLY

LEAD CARD 2
(OMIT IF COLUMN 16 OF LEAD CARD 7 EQUALS 0)

COLUMN		ITEMS	SCALING
1-12	G	= GRAVITATIONAL CONSTANT (FT/SEC**2)	F12.5
13-24	DMM	= BODY MASS (SLUGS)	F12.5
25-36	DIX	= AXIAL MOMENT OF INERTIA (SLUG-FT**2)	F12.5
37-48	DI	= TRANSVERSE MOMENT OF INERTIA OR IY (SLUG-FT**2)	F12.5
49-60	DIZ	= MOMENT OF INERTIA ABOUT Z AXIS (SLUG-FT**2)	F12.5
61-72	DIXY	= PRODUCT OF INERTIA-IXY (SLUG-FT**2)	F12.5

LEAD CARD 3

(OMIT IF COLUMN 17 OF LEAD CARD 7 EQUALS 0)

COLUMN		ITEM	SCALING
1-12	S	= AERODYNAMIC REFERENCE AREA (FT**2)	F12.5
13-24	DEE	= AERODYNAMIC REFERENCE LENGTH (FT)	F12.5
25-36	THETA	= INITIAL EULER ATTITUDE (DEG)	F12.5
37-48	PSI	= INITIAL EULER ATTITUDE (DEG)	F12.5
49-60	Z	= INITIAL VERTICAL COORDINATE (FT)	F12.5
61-72	PHI	= INITIAL EULER ATTITUDE (DEG)	F12.5

Z MUST BE NEGATIVE.

LEAD CARD 4

(OMIT IF 18 OF LEAD CARD 7 EQUALS 0)

COLUMN		ITEM	SCALING
1-12	XDOT	= INITIAL VELOCITY VECTOR (FT/SEC)	F12.5
13-24	YDOT	= INITIAL VELOCITY VECTOR (FT/SEC)	F12.5
25-36	ZDOT	= INITIAL VELOCITY VECTOR (VERTICAL)(FT/SEC)	F12.5
37-48	P	= INITIAL ANGULAR VELOCITY (RAD/SEC)-SPIN	F12.5
49-60	Q	= INITIAL ANGULAR VELOCITY (RAD/SEC)-PITCH	F12.5
61-72	R	= INITIAL ANGULAR VELOCITY (RAD/SEC)-YAW	F12.5

LEAD CARD 5

(OMIT IF COLUMN 19 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM		SCALING
1-12	TSTEP	= COMPUTATION TIME INTERVAL (SEC)	F12.5
13-16	BLANK		
17-24	INCPT1	= PRINT-OUT FREQUENCY (INTERVALS BETWEEN PRINT)	I8
25-36	TMAX	= TIME STOP	F12.5
37-48	ZSTOP	= COORDINATE STOP	F12.5
49-60	TCH	= TIME TO BEGIN HOLDING Q DOT AND R DOT EQUAL TO ZERO AND TO CHANGE INTEGRATION INTERVAL (SEC)	F12.5
61-72	TNEW	= NEW INTEGRATION INTERVAL (SEC)	F12.5
73-76	INCPT2	= PRINT OUT FREQUENCY (INTERVALS BETWEEN PRINT)	I4

INCPT MUST BE GREATER THAN ZERO AND RIGHT ADJUSTED.

LEAD CARD 6

(OMIT IF COLUMN 21 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM		SCALING
1-12	DY	= C.G. OFFSET (FT)	5F12.5
13-24	ZETD1	= FIN ORIENTATION (DEG)	5F12.5
25-36	ZETD2	= WING OR NOSE ORIENTATION (DEG)	5F12.5
37-48	ETA1	= NO. OF FINS (WINGS)	5F12.5
49-60	ETA2	= NO. OF WINGS (FINS)	5F12.5

LEAD CARD 7
(OMIT IF NO CASE FOLLOWS)

COLUMN	ITEM	SCALING
1	KRD(1) = OPTION TO READ IN NEW SUBSCRIPT CARD	I1
2	KRD(2) = OPTION TO READ IN NEW ANGLE OF ATTACK CARD	I1
3	KRD(3) = OPTION TO READ IN NEW MACH VALUE CARD	I1
4	KRD(4) = OPTION TO READ IN NEW SPIN VALUE CARD	I1
5	KRD(5) = OPTION TO READ IN NEW SET OF AXIAL FORCE COEFFICIENTS, CX	I1
6	KRD(6) = OPTION TO READ IN NEW SET OF NORMAL FORCE COEFFICIENTS, CN	I1
7	KRD(7) = OPTION TO READ IN NEW SET OF OVERTURNING MOMENT VARIATIONS, CM	I1
8	KRD(8) = OPTION TO READ IN NEW SET OF DAMPING DERIVATIVES, CMQ	I1
9	KRD(9) = OPTION TO READ IN NEW SET OF DAMPING DERIVATIVES, CNR	I1
10	KRD(10)= OPTION TO READ IN NEW SET OF DAMPING DERIVATIVES, CMPR	I1
11	KRD(11)= OPTION TO READ IN NEW SET OF DAMPING DERIVATIVES, CNPQ	I1
12	KRD(12)= OPTION TO READ IN NEW SET OF MAGNUS FORCES, CNP	I1
13	KRD(13)= OPTION TO READ IN NEW SET OF SPIN COEFFICIENTS, CL	I1
14	KRD(14)= OPTION TO READ IN NEW SET OF SPIN COEFFICIENTS, CLP	I1
15	KRD(15)= OPTION TO READ IN NEW SET OF MAGNUS MOMENTS, CMP	I1

```
16      KRD(16)= OPTION TO READ IN NEW LEAD CARD 2.                 I1
17      KRD(17)= OPTION TO READ IN NEW LEAD CARD 3.                 I1
18      KRD(18)= OPTION TO READ IN NEW LEAD CARD 4.                 I1
19      KRD(19)= OPTION TO READ IN NEW LEAD CARD 5.                 I1
20      KDD(20)= OPTION TO READ IN NEW WIND TABLES.                 I1
21      KDD(21)= OPTION TO READ IN NEW DY, ZET1, ZET2,              I1
                 ETA1, ETA2, EPSMIN.
22      KDD(22)= OPTION TO READ IN NEW CYO.                         I1
23      KDD(23)= OPTION TO READ IN NEW CZO.                         I1
24      KDD(24)= OPTION TO READ IN NEW CMO.                         I1
25      KDD(25)= OPTION TO READ IN NEW CNO.                         I1
26      KDD(26)= OPTION TO READ IN NEW CSF1.                        I1
27      KDD(27)= OPTION TO READ IN NEW CN1.                         I1
28      KDD(28)= OPTION TO READ IN NEW CSF3.                        I1
29      KDD(29)= OPTION TO READ IN NEW CSM1.                        I1
30      KDD(30)= OPTION TO READ IN NEW CM1.                         I1
31      KDD(31)= OPTION TO READ IN NEW CSM3.                        I1
32      KDD(32)= OPTION TO READ IN NEW CLPH1.                       I1
33      KDD(33)= OPTION TO READ IN NEW CLPH2.                       I1

                 = 1 INCLUDE NECESSARY CARD OR CARDS;
                 = 0 OMIT CARD OR CARDS.
```

ANGLE OF ATTACK CARD(S)
(OMIT IF COLUMN 2 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	TABA(I) = ANGLE OF ATTACK (RADIANS) 6 VALUES PER CARD (MAX OF 20 VALUES) I=1,IMAX	6F12.5

MACH VALUE CARD
(OMIT IF COLUMN 3 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-60	TABM(I) = MACH NUMBER. (MAX. OF 5 VALUES) I=1,JMAX	5F12.5

SPIN VALUE CARD
(OMIT IF COLUMN 4 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-60	TABAK(I) = SPIN VALUE. (MAX.OF 5 VALUES) I=1,KMAX	5F12.5

BASIC AEROBALLISTIC COEFFICIENT TABLES

AXIAL FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 5 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CX(I,J) = AXIAL FORCE COEFFICIENT, 6 VALUES PER CARD.MAX.OF 100 VALUES. ((J=1, JMAX), I=1, IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE	6F12.5

NORMAL FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 6 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CN(I,J) = NORMAL FORCE COEFFICIENT. 6 VALUES PER CARD.MAX.OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

OVERTURNING MOMENT VARIATIONS CARD(S)
(OMIT IF COLUMN 7 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CM(I,J) = OVERTURNING MOMENT VARIATION. 6 VALUES PER CARD.MAX.OF 100 VALUES. ((J=1,JMAX),I=1,IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

DAMPING DERIVATIVE CARD(S)
(OMIT IF COLUMN 8 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CMQ(I,J)= DAMPING DERIVATIVE. 6 VALUES PER CARD. MAX. OF 100 VALUES. ((J=1,JMAX),I=1,IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

DAMPING DERIVATIVE CARD(S)
(OMIT IF COLUMN 9 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CNR(I,J)= DAMPING DERIVATIVE. 6 VALUES PER CARD. MAX. OF 100 VALUES. ((J=1,JMAX),I=1,IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

DAMPING DERIVATIVE CARD(S)
(OMIT IF COLUMN 10 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CMPR(I,J)=DAMPING DERIVATIVE. 6 VALUES PER CARD. MAX OF 100 VALUES. ((J=1,JMAX),I=1,IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

DAMPING DERIVATIVE CARD(S)
(OMIT IF COLUMN 11 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CNPQ(I,J)= DAMPING DERIVATIVE. 6 VALUES PER CARD. MAX.OF 100 VALUES. ((J=1,JMAX),I=1,IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

MAGNUS FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 12 OF LEAD CARD 7 EQUALS 0)

```
COLUMN                   ITEM                                          SCALING
 1-72   CNP(I,J,K)=MAGNUS FORCE COEFFICIENT. 6 VALUES PER              6F12.5
                   CARD. MAX. OF 500 VALUES.
                   (((K=1,KMAX),J=1,JMAX),I=1,IMAX)
                   FOR EACH MACH VALUE THERE WILL BE ONE
                   COEFFICIENT FOR EACH SPIN VALUE. THESE FORM
                   A SUBTABLE. THERE WILL BE ONE SUBTABLE FOR
                   EACH ANGLE OF ATTACK VALUE.

                   NOTE- IF BODY FIXED CNP
                         IF FIXED PLANE CLP
                         CNP=-CLP
```

SPIN COEFFICIENT CARD(S)
(OMIT IF COLUMN 13 OF LEAD CARD 7 EQUALS 0)

```
COLUMN                   ITEM                                          SCALING
 1-72   CL(I,J,K)= SPIN COEFFICIENT. 6 VALUES PER CARD.                6F12.5
                   MAX. OF 500 VALUES.
                   (((K=1,KMAX),J=1,JMAX),I=1,IMAX)
                   FOR EACH MACH VALUE THERE WILL BE ONE
                   COEFFICIENT FOR EACH SPIN VALUE. THESE FORM
                   A SUBTABLE. THERE WILL BE ONE SUBTABLE FOR
                   EACH ANGLE OF ATTACK VALUE.
```

SPIN COEFFICIENT CARD(S)
(OMIT IF COLUMN 14 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CLP(I,J,K)= SPIN COEFFICIENT. 6 VALUES PER CARD. MAX. OF 500 VALUES. (((K=1,KMAX),J=1,JMAX),I=1,IMAX) FOR EACH MACH VALUE THERE WILL BE ONE COEFFICIENT FOR EACH SPIN VALUE. THESE FORM A SUBTABLE. THERE WILL BE ONE SUBTABLE FOR EACH ANGLE OF ATTACK VALUE.	6F12.5

MAGNUS MOMENT CARD(S)
(OMIT IF COLUMN 15 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CMP(I,J,K)=MAGNUS MOMENT. 6 VALUES PER CARD. MAX. OF 500 VALUES. (((K=1,KMAX),J=1,JMAX),I=1,IMAX) FOR EACH MACH VALUE THERE WILL BE ONE COEFFICIENT FOR EACH SPIN VALUE. THESE FORM A SUBTABLE. THERE WILL BE ONE SUBTABLE FOR EACH ANGLE OF ATTACK VALUE.	6F12.5

WIND TABLES (OPTIONAL)
(OMIT IF COLUMN 20 OF LEAD CARD 7 EQUALS 0)

(6 VALUES PER CARD, MAXIMUM OF 10)

COLUMN	ITEM	SCALING
1-72	TABZ(I) = VERTICAL COORDINATE, FT	6F12.5
1-72	WDX(I) = WIND VELOCITY IN DIRECTION OF X COORDINATE, FT/SEC	6F12.5
1-72	WDY(I) = WIND VELOCITY IN DIRECTION OF Y COORDINATE, FT/SEC	6F12.5

TABZ MUST BE IN POSITIVE ASCENDING ORDER-ALTITUDE CAN BE USED INSTEAD OF Z COORDINATE IF DESIRED

MAXIMUM NUMBER OF WIND VALUES (VERTICAL COORDINATE) IS TEN

ADDITIONAL AERODYNAMIC COEFFICIENT TABLES

TRIM FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 22 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CYO(I,J) = TRIM FORCE COEFFICIENT, Y BODY FIXED AXIS, 6 VALUES PER CARD.MAX.OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

TRIM FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 23 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CZO(I,J) = TRIM FORCE COEFFICIENT, Z BODY FIXED AXIS, 6VALUES PER CARD,MAX,OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

TRIM MOMENT COEFFICIENT CARD(S)
(OMIT IF COLUMN 24 OF LEAD CARD 7 EQUALS 0)

```
COLUMN               ITEM                                       SCALING
 1-72   CMO(I,J) = TRIM MOMENT COEFFICIENT, ABOUT Y BODY-FIXED    6F12.5
                   AXIS, 6 VALUES PER CARD.MAX.OF 100 VALUES.
                   ((J=1,JMAX),I=1=IMAX)
                   FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE
                   ONE COEFFICIENT FOR EACH MACH VALUE.
```

TRIM MOMENT COEFFICIENT CARD(S)
(OMIT IF COLUMN 25 OF LEAD CARD 7 EQUALS 0)

```
COLUMN               ITEM                                       SCALING
 1-72   CNO(I,J) = TRIM MOMENT COEFFICIENT, ABOUT Z BODY-FIXED    6F12.5
                   AXIS, 6 VALUES PER CARD.MAX OF 100 VALUES.
                   ((J=1,JMAX),I=1=IMAX)
                   FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE
                   ONE COEFFICIENT FOR EACH MACH VALUE.
```

SIDE FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 26 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CSF1(I,J)= SIDE FORCE COEFFICIENT DUE TO AERODYNAMIC ROLL ANGLE OF FINS, 6 VALUES PER CARD. MAX.OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

SIDE FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 27 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CN1(I,J)= NORMAL FORCE COEFFICIENT DUE TO AERODYNAMIC ROLL ANGLE OF FINS (WINGS), 6 VALUES PER CARD. MAX. OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

SIDE FORCE COEFFICIENT CARD(S)
(OMIT IF COLUMN 28 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CSF3(I,J)= SIDE FORCE COEFFICIENT DUE TO ASYMMETRIC VORTICES, 6 VALUES PER CARD. MAX. OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

SIDE MOMENT COEFFICIENT CARD(S)
(OMIT IF COLUMN 29 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CSM1(I,J)= SIDE MOMENT COEFFICIENT DUE TO AERODYNAMIC ROLL ANGLE OF FINS, 6 VALUES PER CARD. MAX. OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

SIDE MOMENT COEFFICIENT CARD(S)
(OMIT IF COLUMN 30 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CM1(I,J)=PITCH MOMENT COEFFICIENT DUE TO AERODYNAMIC ROLL ANGLE OF FINS (WINGS), 6 VALUES PER CARD.MAX.OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

SIDE MOMENT COEFFICIENT CARD(S)
(OMIT IF COLUMN 31 OF LEAD CARD 7 EQUALS 0)

COLUMN	ITEM	SCALING
1-72	CSM3(I,J)= SIDE MOMENT COEFFICIENT DUE TO ASYMMETRIC VORTICES, 6 VALUES PER CARD. MAX OF 100 VALUES. ((J=1,JMAX),I=1=IMAX) FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE ONE COEFFICIENT FOR EACH MACH VALUE.	6F12.5

```
                 ROLLING MOMENT COEFFICIENT CARD(S)
            (OMIT IF COLUMN 32 OF LEAD CARD 7 EQUALS 0)

COLUMN                ITEM                                      SCALING
 1-72 CLPH1(I,J)= ROLLING MOMENT COEFFICIENT DUE TO              6F12.5
                  AERODYNAMIC ROLL ANGLE OF FINS. 6 VALUES
                  PER CARD. MAX.OF 100 VALUES.
                  ((J=1,JMAX),I=1=IMAX)
                  FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE
                  ONE COEFFICIENT FOR EACH MACH VALUE.


                 ROLLING MOMENT COEFFICIENT CARD(S)
            (OMIT IF COLUMN 33 OF LEAD CARD 7 EQUALS 0)

COLUMN                ITEM                                      SCALING
 1-72 CLPH2(I,J)= ROLLING MOMENT COEFFICIENT DUE TO              6F12.5
                  AERODYNAMIC ROLL ANGLE OF WINGS,
                  6 VALUES PER CARD.MAX.OF 100 VALUES.
                  ((J=1,JMAX)I=1=IMAX)
                  FOR EACH ANGLE OF ATTACK VALUE THERE WILL BE
                  ONE COEFFICIENT FOR EACH MACH VALUE.
```

SECTION III

OUTPUT FORMAT

SYSTEM OUTPUT (B1-6)

LINE 1

WORD	ITEM		SCALING
1	TIME	= PRINT TIME (SEC)	F7.4
2	CAPV	= TOTAL VELOCITY (FT/SEC)	F12.1
3	Y(7)	= X RANGE COORDINATE (FT)	F12.1
4	Y(8)	= Y RANGE COORDINATE (FT)	F12.1
5	Y(9)	= Z COORDINATE, ALTITUDE (FT)	F12.1
6	Z(7)	= VELOCITY IN DIRECTION OF X RANGE COORDINATE (FT/SEC)	F12.1
7	Z(8)	= VELOCITY IN DIRECTION OF Y RANGE COORDINATE (FT/SEC)	F12.1
8	Z(9)	= VERTICAL VELOCITY (FT/SEC)	F12.1
9	ALD	= ANGLE OF ATTACK (DEG)	F7.2
10	CP1	= FIN OFFSET ANGLE (DEG)	F7.2
11	CP2	= WING OR NOSE OFFSET ANGLE (DEG)	F7.2
12	CAPVA	= AERODYNAMIC VELOCITY	F10.1
13	EM	= MACH NUMBER	F7.2

LINE 2

WORD	ITEM		SCALING
1	Y(1)	= VELOCITY IN DIRECTION OF X MOVING COORDINATE-BODY OR FIXED PLANE AXES (FT/SEC)	F12.1
2	Y(2)	= VELOCITY IN DIRECTION OF Y MOVING COORDINATE-BODY OR FIXED PLANE AXES (FT/SEC)	F12.1
			F12.2
3	Y(3)	= VELOCITY IN DIRECTION OF Z MOVING COORDINATE-BODY OR FIXED PLANE AXES (FT/SEC)	F12.1
4	Y(4)	= SPIN RATE (RAD/SEC)	
5	Y(5)	= PITCH RATE (RAD/SEC)	F12.2
6	Y(6)	= YAW RATE (RAD/SEC)	F12.2
7	PHD	= ROLL ATTITUDE (DEG)	F11.2
8	THD	= PITCH ATTITUDE (DEG)	F11.2
9	PSD	= YAW ATTITUDE (DEG)	F11.2
10	EPSIL	= INTEGRATION ERROR IN QUATERNIONS	1PE17.6

SYSTEM OPTIONAL OUTPUT (A6-26)

LINE 1

WORD	ITEM		SCALING
1	TIME	= PRINT TIME (SEC)	F9.4
2	PHD	= ROLL ATTITUDE RATE (RAD/SEC)	F15.2
3	THD	= PITCH ATTITUDE RATE (RAD/SEC)	F15.2
4	PSD	= YAW ATTITUDE RATE (RAD/SEC)	F15.2
5	Y(10)	= QUATERNION VALUE	F15.6
6	Y(11)	= QUATERNION VALUE	F15.6
7	Y(12)	= QUATERNION VALUE	F15.6
8	Y(13)	= QUATERNION VALUE	F15.6

LINE 2

WORD	ITEM		SCALING
1	E(4)	= ERROR IN SPIN RATE (RAD/SEC)	1PE15.6
2	E(5)	= ERROR IN PITCH RATE (RAD/SEC)	E15.6
3	E(6)	= ERROR IN YAW RATE (RAD/SEC)	E15.6
4	PDV	= NON-DIMENSIONAL SPIN VALUE	E15.6
5	E(10)	= QUATERNION ERROR (ACCUMULATIVE)	E15.6
6	E(11)	= QUATERNION ERROR (ACCUMULATIVE)	E15.6
7	E(12)	= QUATERNION ERROR (ACCUMULATIVE)	E15.6
8	E(13)	= QUATERNION ERROR (ACCUMULATIVE)	E15.6

SECTION IV

PROGRAM FLOW CHARTS AND SUBROUTINE DESCRIPTION

Description of Labeled Common Names

Name	Description	Reference Section	Reference Page
⟨BTOF⟩	Arguments of SETMAX		
XMAT(3, 3)	Rotational Matrix	I	4
⟨MPGIN⟩	Input Related Only to Main Program		
XDOT(3)	$\dot{X}, \dot{Y}, \dot{Z}$	II	31
P	p	II	31
Q	q	II	31
R	r	II	31
ZALT	Z	II	31
PHI	φ	II	31
THETA	θ	II	31
PSI	ψ	II	31
TSTEP	Δt	II	32
INCPT1	Print cycle before $\dot{q} = \dot{r} = 0$	II	32
TMAX	t_{max}	II	32
ZSTOP	Z_{min}	II	32
TCH	t_{change} $(\dot{q} = \dot{r} = 0)$	II	32
TNEW	Δt_2	II	32
NRK	Integration code	II	29
NCMAX	Number of corrections	II	29
EPSMAX	ϵ_{max}	II	29
ZETD1	ζ_1, deg.	II	32
ZETD2	ζ_2, deg.	II	32
INCPT2	Print cycle after $\dot{q} = \dot{r} = 0$	II	32

Description of Labeled Common Names (Continued)

Name	Description	Reference Section	Page
⟨ALLIN⟩	Input Related to Both Main Program and Equations of Motion		
DIX	I_x	II	30
DI	I or I_y	II	30
DIZ	I_z	II	30
DIXY	I_{xy}	II	30
BODFIX	T/F Flag False if Fixed Plane	II	30
⟨EQMPG⟩	Main Program Constants Needed In Equations of Motion		
QR	T/F Flag - if false, use simplified equations for q and r		
RAT	I_x/I_y	I	9
BODJX	I_{xy}/I_x	I	8
BODJY	I_{xy}/I_y	I	8
BODJZ	I_{xy}/I_z	I	8
BODEN	$1 - I_{xy}{}^2/I_x I_y$	I	8
BODJ	$(I_x + I_y - I_z) I_{xy}/(I_x I_y)$	I	8
BODPD	$\left[I_{xy}{}^2 + I_y (I_y - I_z)\right]/(I_x I_y)$	I	8
BODQD	$\left[I_{xy}{}^2 - I_x (I_x - I_z)\right]/(I_x I_y)$	I	8
BODRD	$(I_x - I_y)/I_z$	I	8
ZET1	ζ_1 radians	I	21
ZET2	ζ_2 radians	I	21

Description of Labeled Common Names (Continued)

Name	Description	Reference Section	Page
⟨EQMIN⟩	Input Related Only to Equations of Motion		
IMAX	≤ 20	II	29
JMAX	≤ 5	II	29
KMAX	≤ 5	II	29
LMAX	≤ 10	II	29
TABA (20)	(α)	II	35
TABM (5)	(M)	II	35
TABAR (5)	(pd/2V)	II	35
TABZ (10)	(- Z)	II	41
CX (20, 5)	C_x	II	36
CN (20, 5)	C_N	II	36
CM (20, 5)	C_M	II	36
CMQ (20, 5)	C_{m_q}	II	37
CNR (20, 5)	C_{n_r}	II	37
CMPR (20, 5)	$C_{m_{pr}}$	II	38
CNPQ (20, 5)	$C_{n_{pq}}$	II	38
CNPA (20, 5, 5)	$C_{N_p} = -C_{L_p}$	II	39
CL (20, 5, 5)	C_ℓ	II	39
CLP (20, 5, 5)	C_{ℓ_p}	II	40
CMPA (20, 5, 5)	C_{M_p}	II	40
CYO (20, 5)	C_{y_o}	II	42
CZO (20, 5)	C_{z_o}	II	42
CMO (20, 5)	C_{m_o}	II	43
CNO (20, 5)	C_{n_o}	II	43
CSF1 (20, 5)	C_{SF_1}	II	44
CN1 (20, 5)	C_{N_1}	II	44
CSF3 (20, 5)	C_{SF_3}	II	44

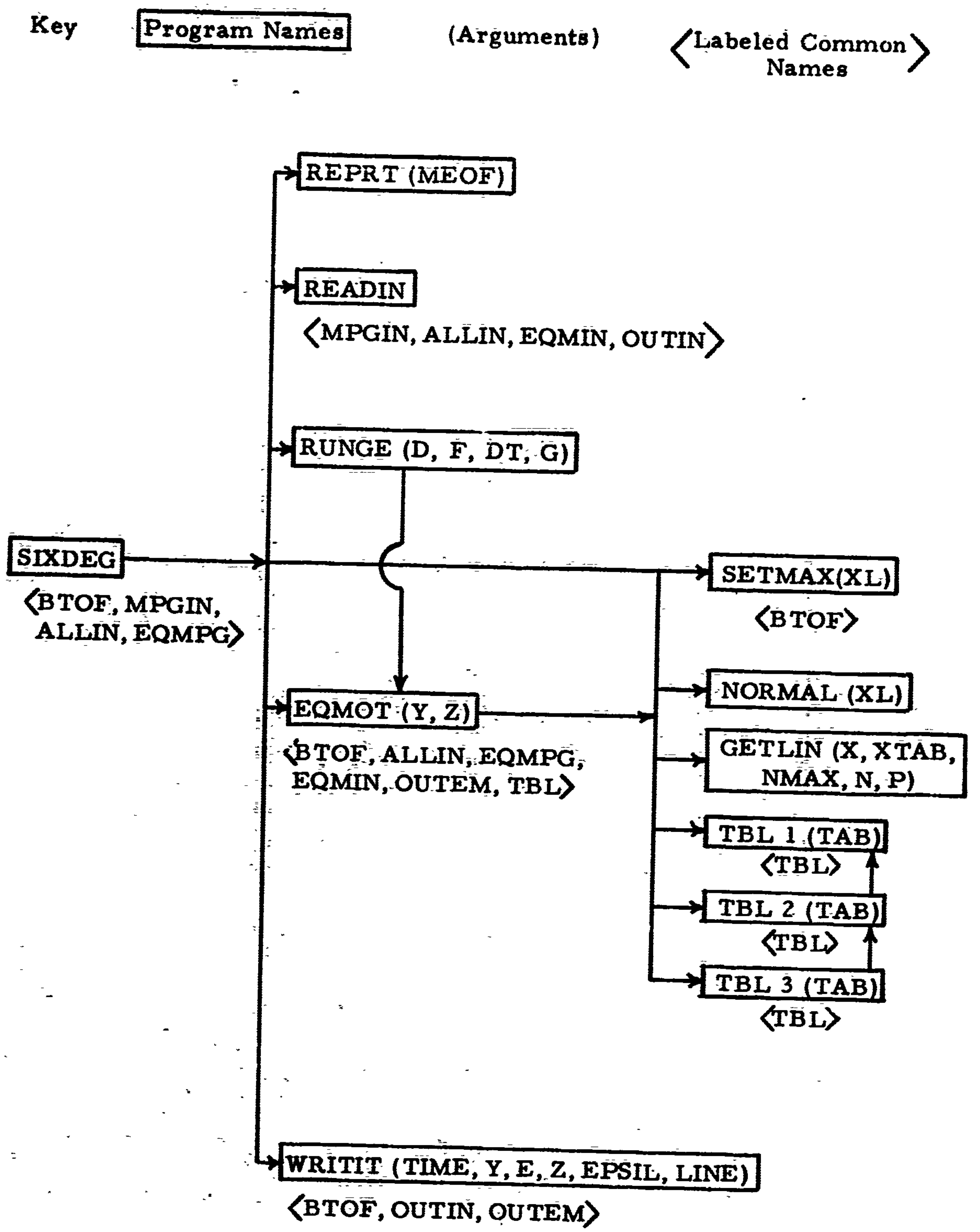
Description of Labeled Common Names (Continued)

Name	Description	Reference Section	Page
⟨EQMIN⟩ (continued)			
CSM1 (20, 5)	C_{SM_1}	II	45
CM1 (20, 5)	C_{M_1}	II	45
CSM3 (20, 5)	C_{SM_3}	II	45
CLPH1 (20, 5)	$C_{\ell \Phi_1}$	II	46
CLPH2 (20, 5)	$C_{\ell \Phi_2}$	II	46
WDX (10)	$\dot{X}_w$	II	41
WDY (10)	$\dot{Y}_w$	II	41
G	g	II	30
DMM	m	II	30
S	S	II	31
DEE	d	II	31
DY	Δy	II	32
ETA1	η_{t1}	II	32
ETA2	η_{t2}	II	32
⟨OUTIN⟩	Input Related Only to Output		
HEADER (11)		II	30
IRUN		II	30
IDATE		II	30
KA6		II	30

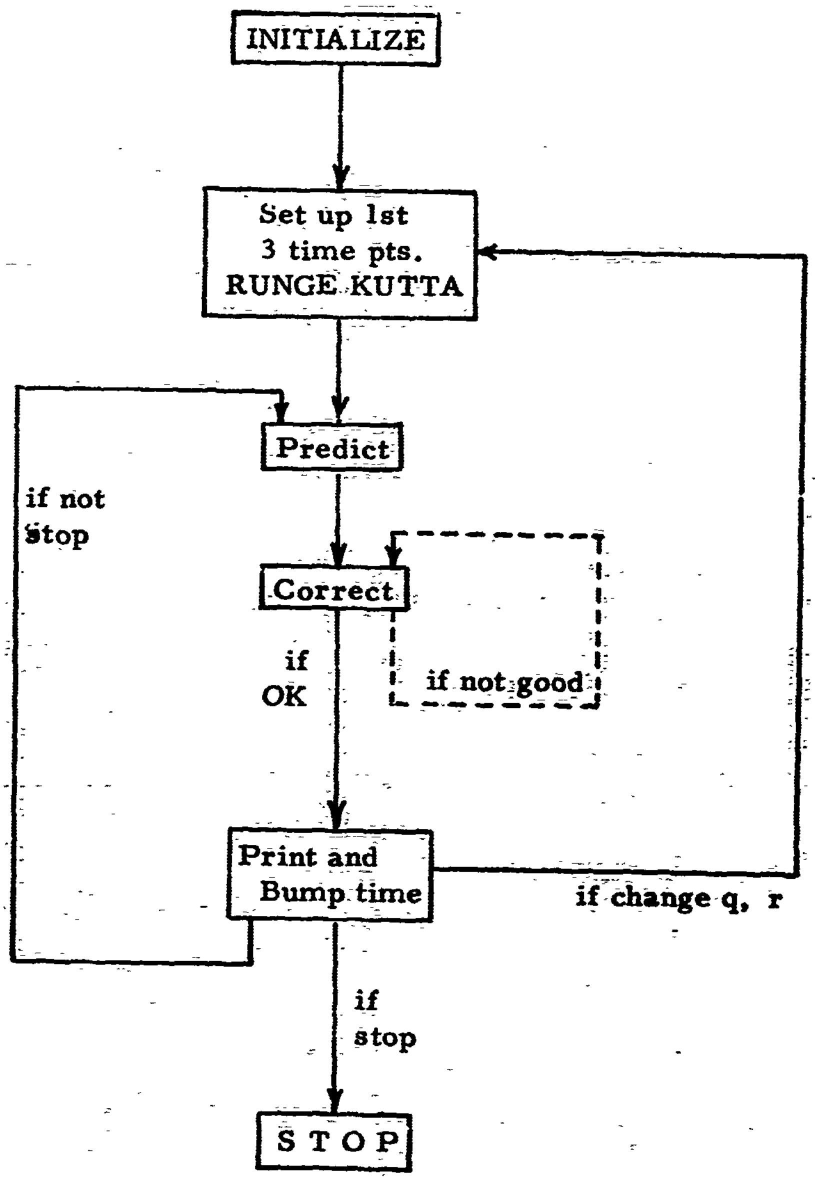
Description of Labeled Common Names (Continued)

Name	Description	Reference Section	Page
⟨TBL⟩	Arguments for TBL1, TBL2, TBL3 found from GETLIN		
N1	n_1 — index such that $XT\,(n-1) < X \leq XT\,(n)$		
N2	n_2 — for $n = n_1$ or n_2 or n_3		
N3	n_3		
P1	1 — ratio of $\dfrac{XT\,(n) - X}{XT\,(n) - XT\,(n-1)}$		
P2	2		
P3	3 — for $n = n_1$ or n_2 or n_3		
⟨OUTEM⟩	Equations of Motion Generated Data for Output		
CAPHI1	$\Phi_1 = \pi/2 - (\xi - \xi_1)$	I	16
CAPHI2	$\Phi_2 = \pi/2 - (\xi - \xi_2)$	I	16
PDV	pd/2V	I	10
ALPHA	$\tilde{\alpha}$	I	10
CAPVA	V_A	I	20
EM	Mach Number	I	10
PEE	p or $(-r \tan \theta)$	I	7, 25

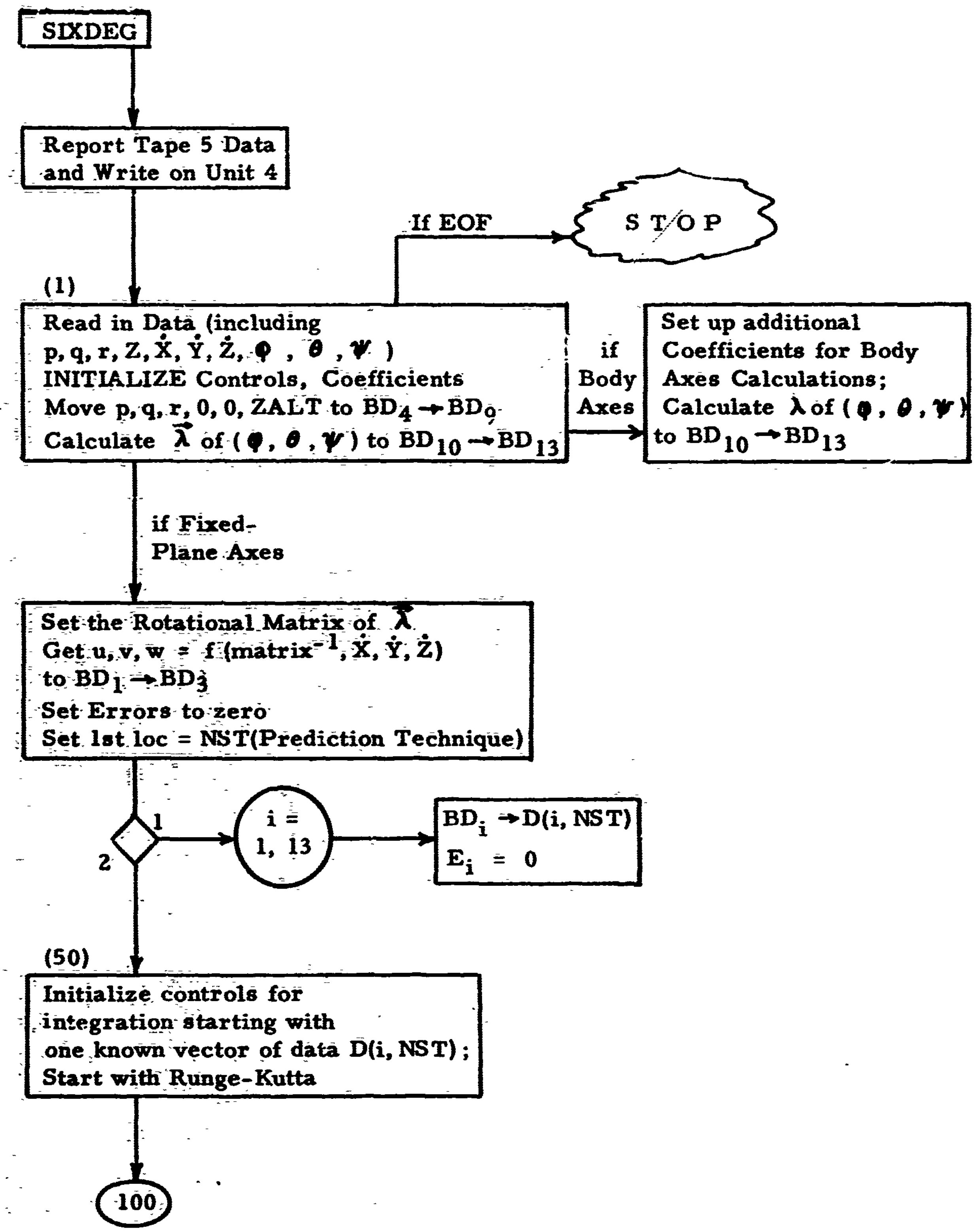
Hierachy of Programs and their Labeled Common



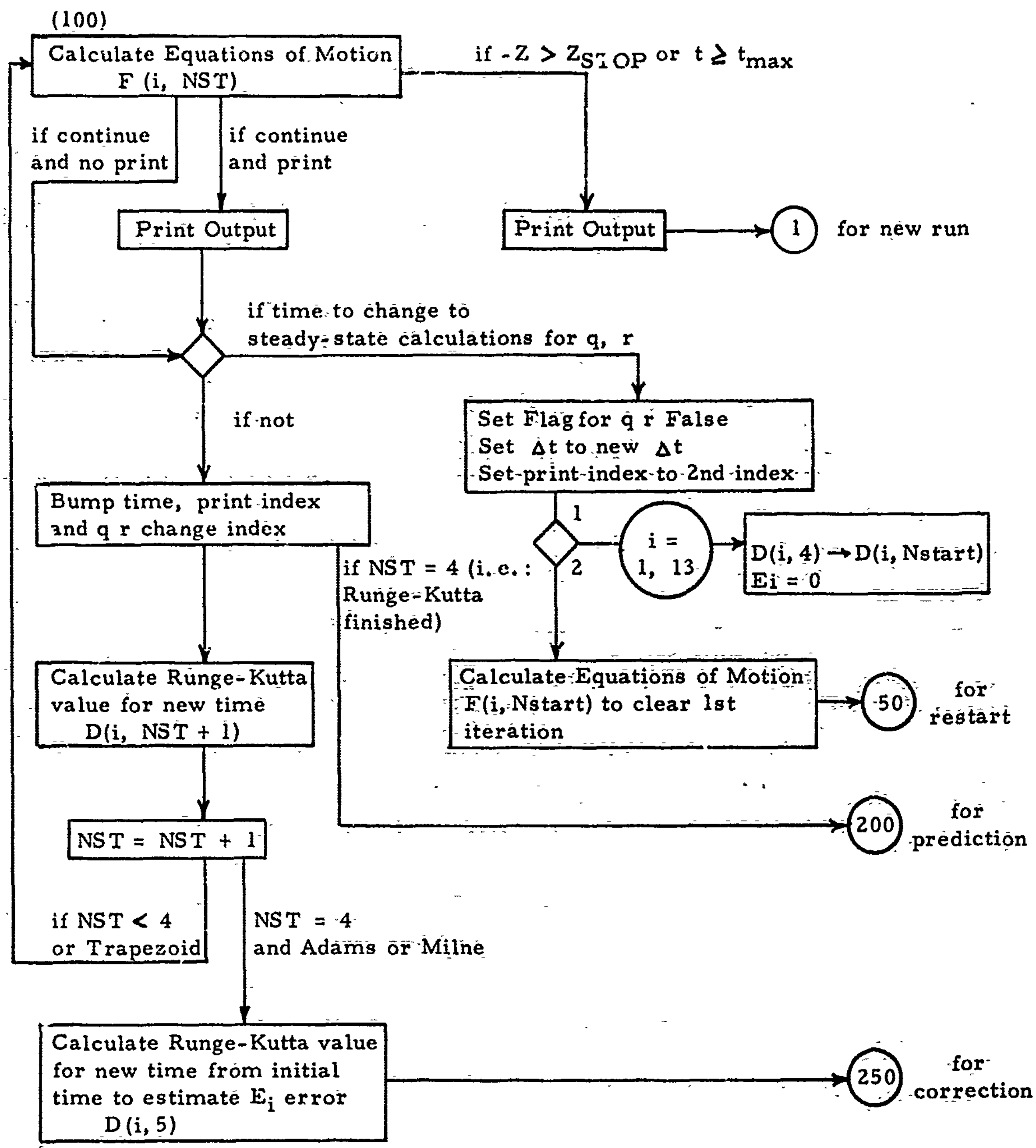
Simple Flow Chart of LOGIC for Single Run



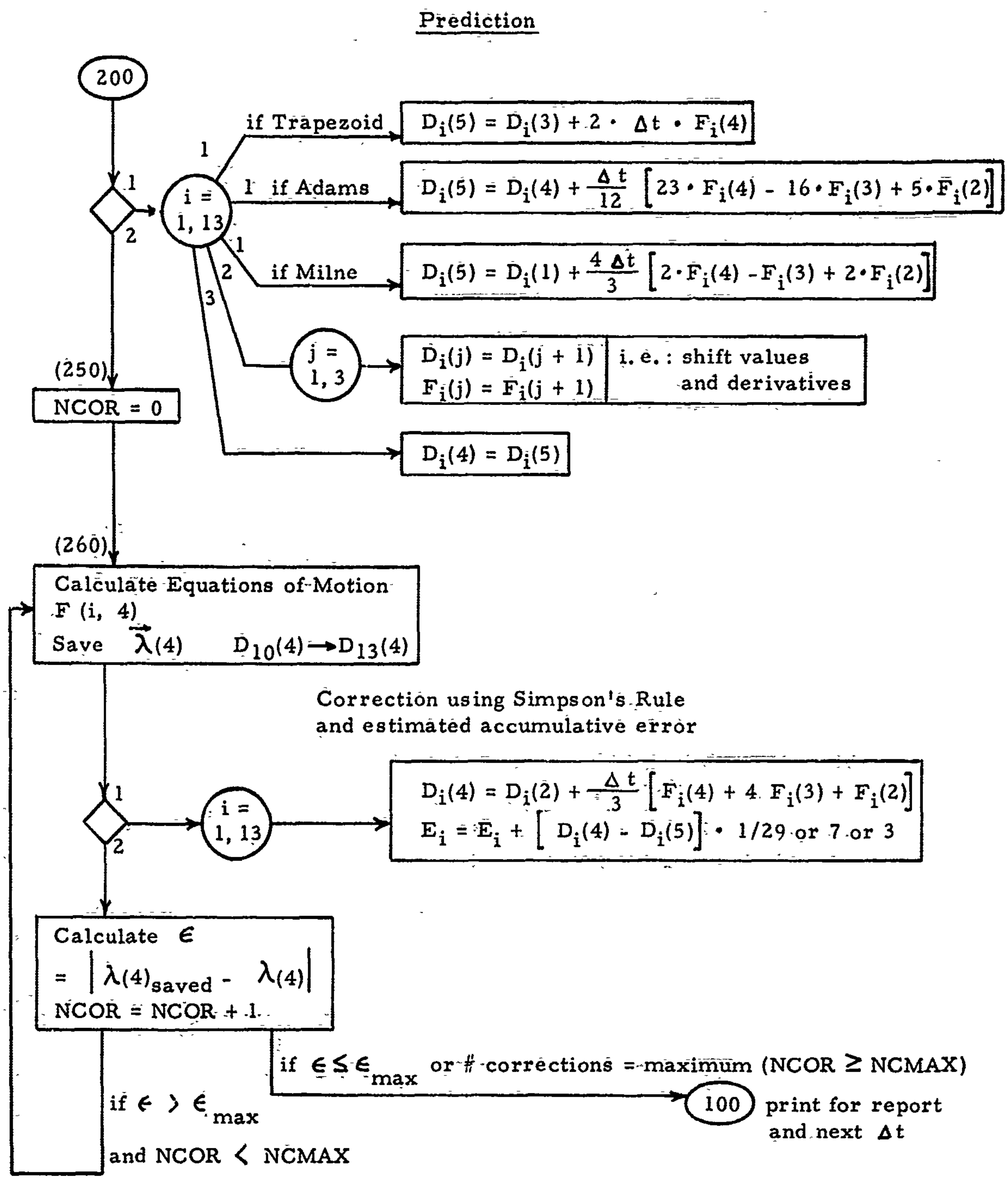
Flow Chart of Main Program



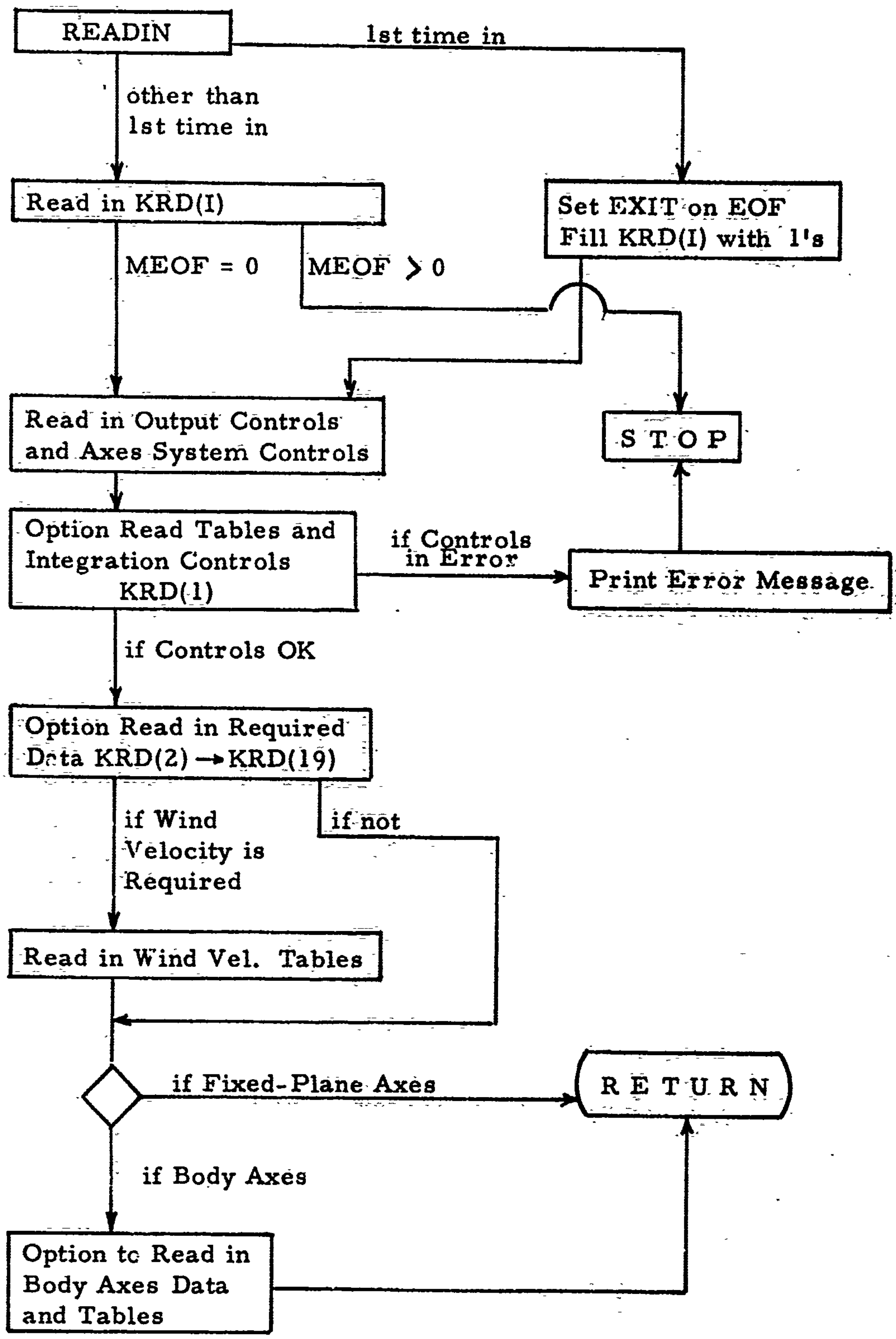
Main Program (continued)



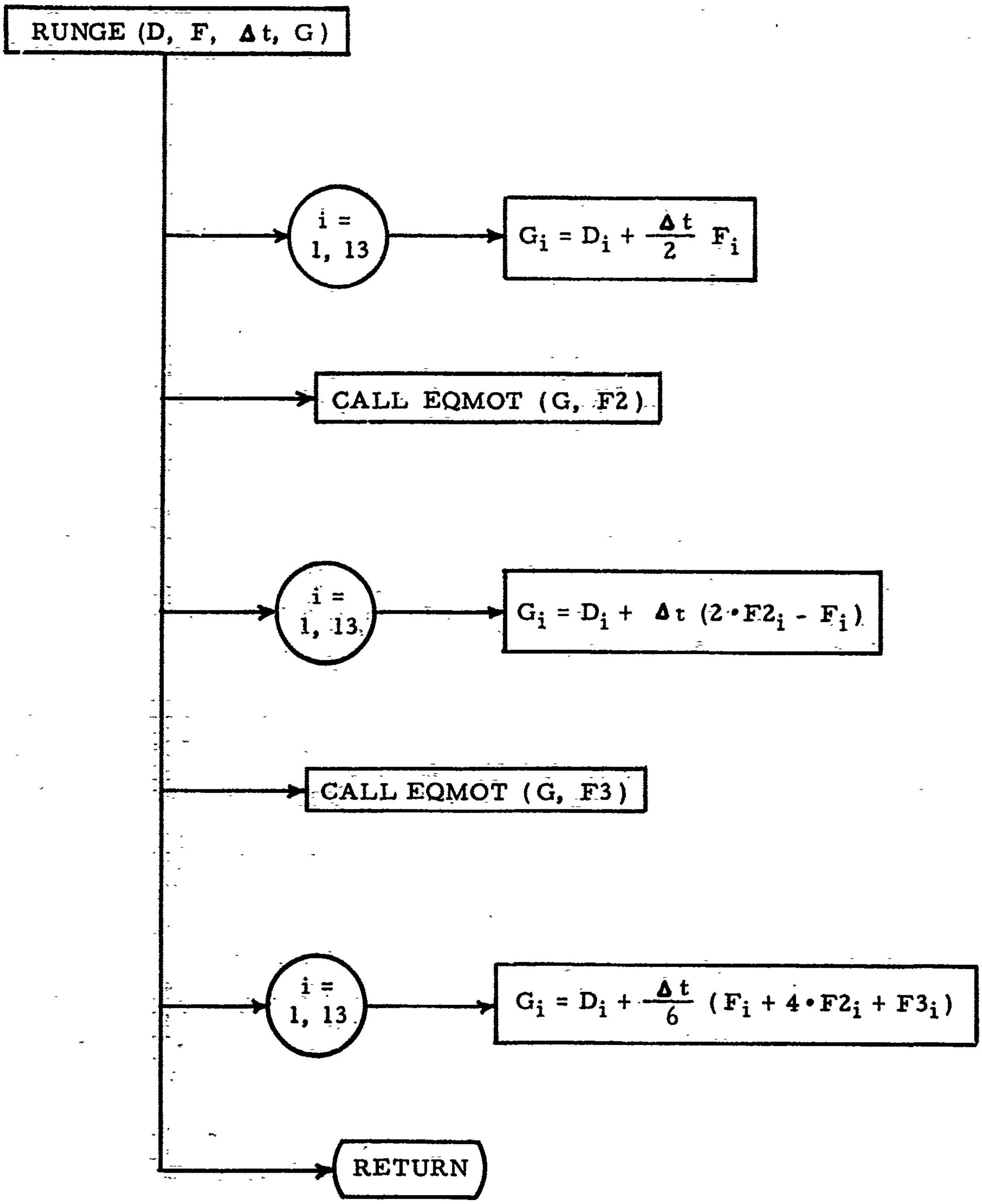
Main Program (concluded)



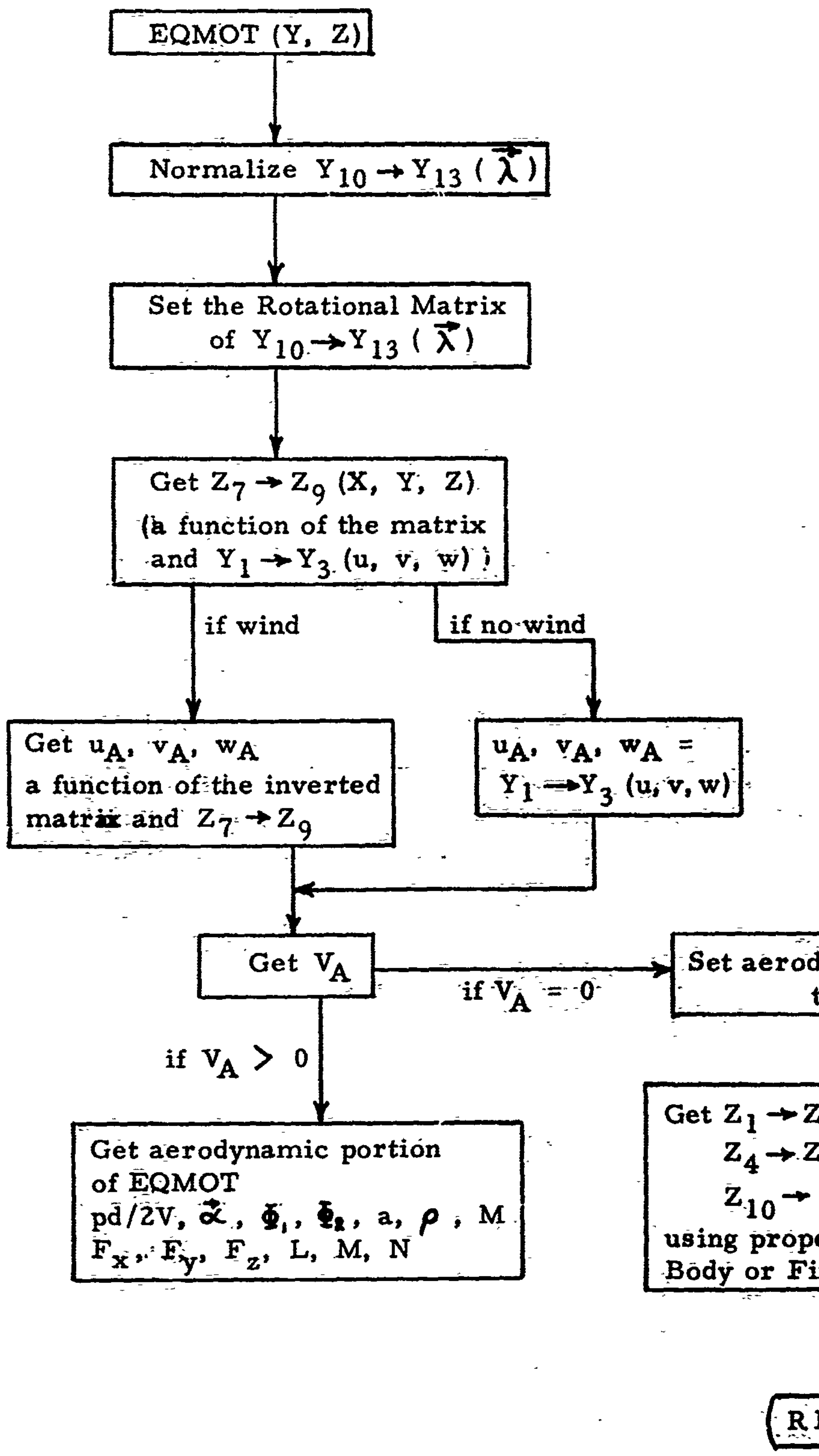
Flow Chart of READIN



Flow Chart of RUNGE



Flow Chart of EQMOT



Y =	Z =
u	$\dot{u}$
v	$\dot{v}$
w	$\dot{w}$
p	$\dot{p}$
q	$\dot{q}$
r	$\dot{r}$
X	$\dot{X}$
Y	$\dot{Y}$
Z	$\dot{Z}$
λ_0	$\dot{\lambda}_0$
λ_1	$\dot{\lambda}_1$
λ_2	$\dot{\lambda}_2$
λ_3	$\dot{\lambda}_3$

$$A = [\, q \sin \xi - r \cos \xi \,] \left(\frac{d}{2V} \right)$$

$$B = [\, q \cos \xi + r \sin \xi \,] \left(\frac{d}{2V} \right)$$

$$C_M ADD = C_M(\vec{\alpha}, M) + \boxed{C_{M_1}(\vec{\alpha}, M) \sin \eta_1 \left[\frac{\pi}{2} - \xi + \xi_1\right]} + C_{m_q}(\vec{\alpha}, M)$$

$$+ B \cdot C_{m_{p_r}}(\vec{\alpha}, M) \cdot \frac{pd}{2V}$$

$$C_{M_p} ADD = C_{M_p}\left(\vec{\alpha}, M, \frac{pd}{2V}\right) \cdot \frac{pd}{2V} + B \cdot C_{n_r}(\vec{\alpha}, M) + A \cdot C_{n_{p_q}}(\vec{\alpha}, M) \cdot \frac{pd}{2V}$$

$$+ \boxed{C_{S_{M_1}}(\vec{\alpha}, M) \sin \left(\eta_1 \left[\frac{\pi}{2} - \xi + \xi_1\right]\right) + C_{S_{M_3}}(\vec{\alpha}, M)}$$

$$M = CON \cdot d^* \left\{ \cancel{\boxed{C_{m_o}(\vec{\alpha}, M)}} + C_M ADD \cdot \sin \xi + C_{M_p} ADD \cdot \cos \xi \right\}$$

$$N = CON \cdot d^* \left\{ \boxed{C_{n_o}(\vec{\alpha}, M)} - C_M ADD \cdot \cos \xi + C_{M_p} ADD \cdot \sin \xi \right\}$$

$$+ \boxed{\Delta_y \cdot F_x}$$

EQMOT Equations in More Detail for F_x, F_y, F_z, L, M, N

$\boxed{}$ denotes body-fixed axes option only

$$CON = \frac{1}{2}\,\rho\, V^2 S$$

$$F_x = CON^* \, C_x(\vec{\alpha}, M)$$

$$C_{N_p}ADD = \boxed{\mp}\; C_{N_p}\left(\vec{\alpha}, M, \frac{pd}{2V}\right)\cdot\frac{pd}{2V}$$

$$+\; C_{SF_1}(\vec{\alpha}, M)\sin\left(\eta_1\left[\frac{\pi}{2} - \xi + \delta_1\right]\right) + C_{SF_3}(\vec{\alpha}, M)$$

$$F_y = CON^* \left\{ \boxed{C_{y_o}(\vec{\alpha}, M)} - \left[C_N(\vec{\alpha}, M) + \boxed{C_{N_1}(\vec{\alpha}, M)\sin\eta_1\left[\frac{\pi}{2} - \xi + \delta_1\right]}\right]\cos\xi \right.$$

$$\left. + \; C_{N_p}ADD \cdot \sin\xi \right\}$$

$$F_z = CON^* \left\{ \boxed{C_{z_o}(\vec{\alpha}, M)} - \left[C_N(\vec{\alpha}, M) + \boxed{C_{N_1}(\vec{\alpha}, M)\sin\eta_1\left[\frac{\pi}{2} - \xi + \delta_1\right]}\right]\sin\xi \right.$$

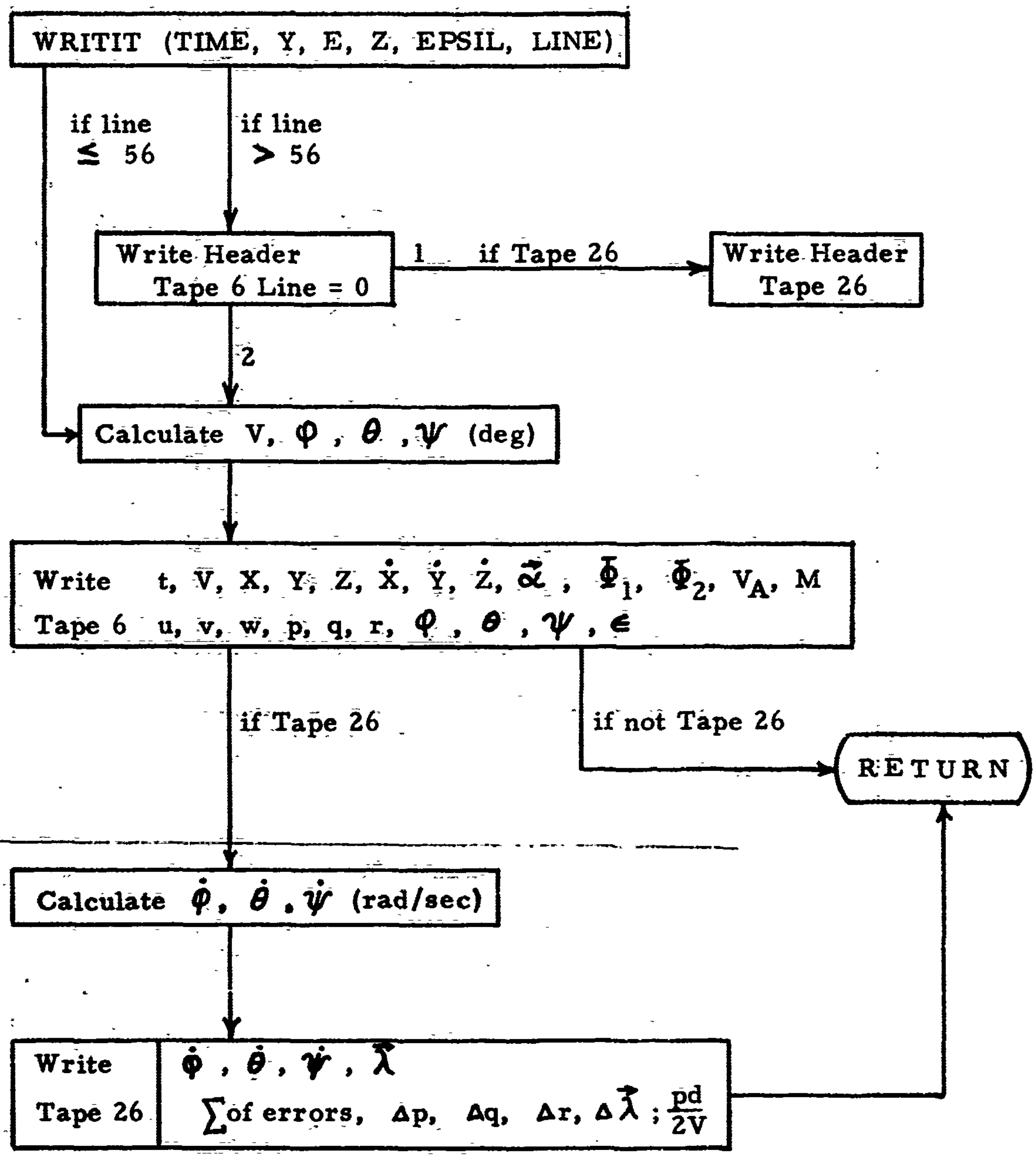
$$\left. - \; C_{N_p}ADD \cdot \cos\xi \right\}$$

$$L = CON\; d^* \left\{ C_{\ell}\left(\vec{\alpha}, M, \frac{pd}{2V}\right) + \frac{pd}{2V}\cdot C_{\ell_p}\left(\vec{\alpha}, M, \frac{pd}{2V}\right) \right.$$

$$\left. + \;\boxed{C_{\ell_{\delta_1}}(\vec{\alpha}, M)\sin\left(\eta_1\left[\frac{\pi}{2} - \xi + \delta_1\right]\right) + C_{\ell_{\delta_2}}(\vec{\alpha}, M)\sin\left(\eta_2\left[\frac{\pi}{2} - \xi + \delta_2\right]\right)} \right\}$$

$$- \;\boxed{\Delta_y \quad F_z}$$

Flow Chart of WRITIT



Flow Chart of SETMAX

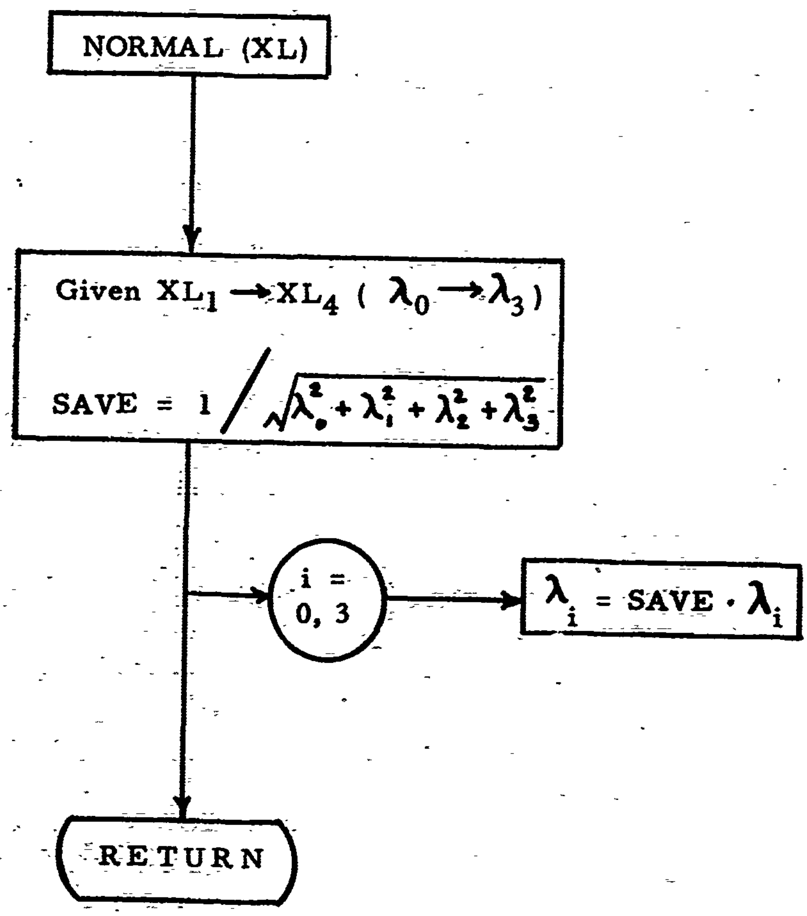
SETMAX (XL)

Given $XL_1 \rightarrow XL_4$ $(\lambda_0 \rightarrow \lambda_3)$

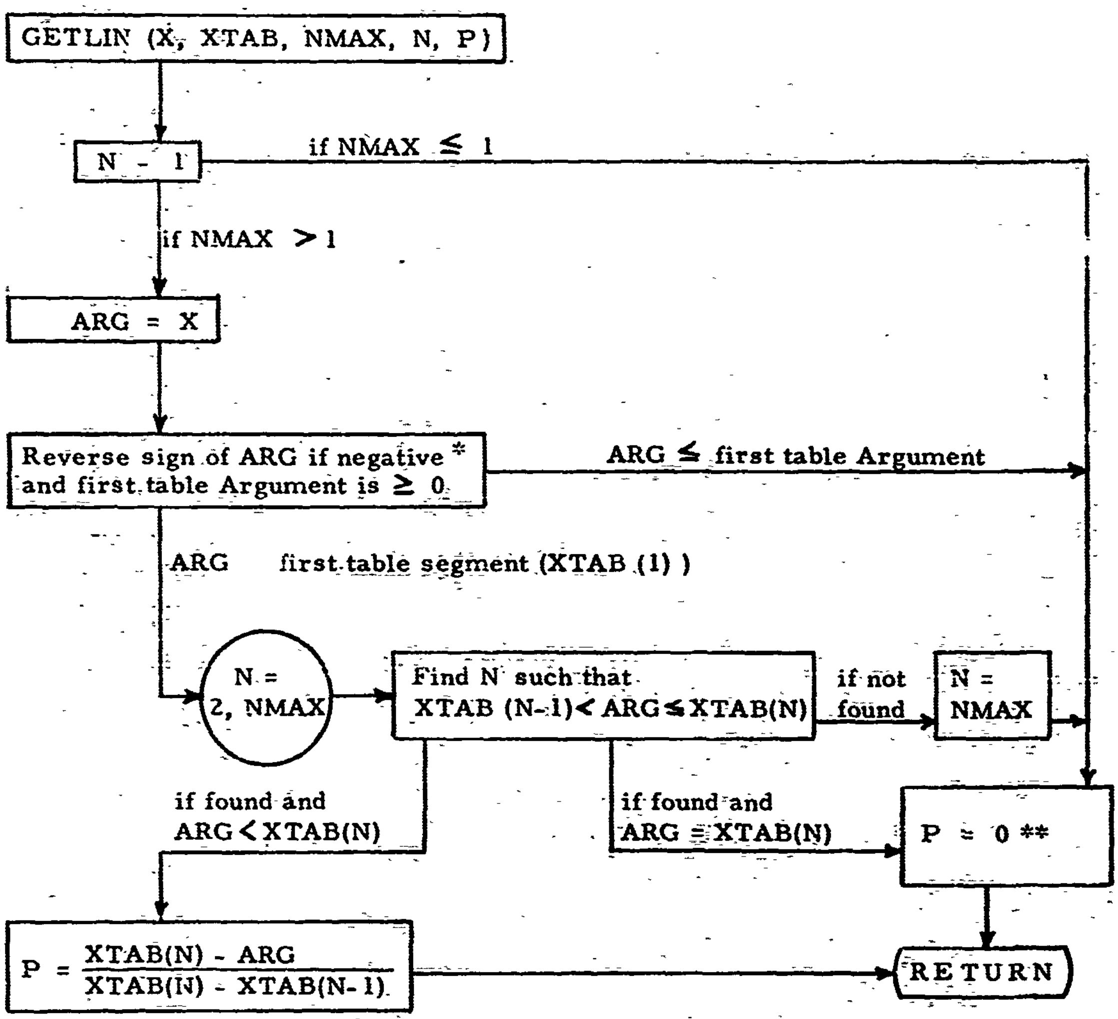
$$\begin{aligned}
XMAT(1,1) &= \lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2 \\
XMAT(2,2) &= \lambda_0^2 - \lambda_1^2 + \lambda_2^2 - \lambda_3^2 \\
XMAT(3,3) &= \lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2 \\
XMAT(1,2) &= 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) \\
XMAT(2,1) &= 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) \\
XMAT(1,3) &= 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\
XMAT(3,1) &= 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) \\
XMAT(2,3) &= 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\
XMAT(3,2) &= 2(\lambda_2\lambda_3 + \lambda_0\lambda_1)
\end{aligned}$$

RETURN

Flow Chart of NORMAL



Flow Chart of GETLIN



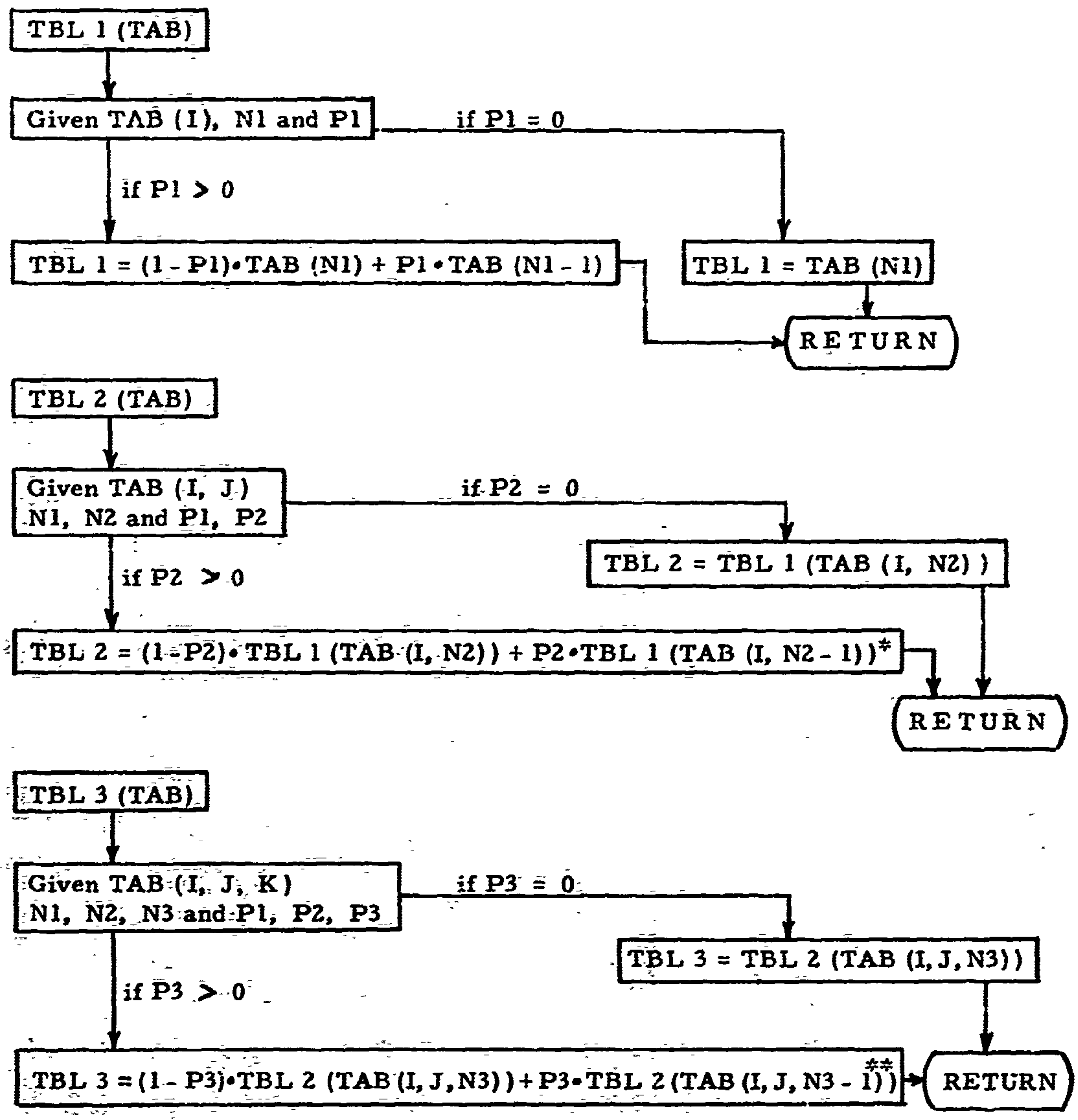
Note: N & P are used such that

$$Y = (1 - P) \cdot Y_N + P \cdot Y_{N-1} \quad \text{in TBL 1, 2 and 3}$$

* i. e., to handle negative altitude and negative p where only + arguments are in Table.

** i. e., saves interpolation.

Flow Chart of TBL 1, TBL 2, TBL 3



$$* \quad TBL\ 2 = (1-P2)\left[(1-P1)\cdot TAB\ (N1, N2) + P1\cdot TAB\ (N1-1, N2)\right] + P2\left[(1-P1)\cdot TAB\ (N1, N2-1) + P1\cdot TAB\ (N1-1, N2-1)\right]$$

$$** \quad TBL\ 3 = (1-P3)\Big[(1-P2)\left\{(1-P1)\cdot TAB\ (N1, N2, N3) + P1\cdot TAB(N1-1, N2, N3)\right\} + P2\left\{(1-P1)\cdot TAB\ (N1, N2-1, N3) + P1\cdot TAB\ (N1-1, N2-1, N3)\right\}\Big] + P3\Big[(1-P2)\left\{(1-P1)\cdot TAB\ (N1, N2, N3-1) + P1\cdot TAB\ (N1-1, N2, N3-1)\right\} + P2\left\{(1-P1)\cdot TAB\ (N1, N2-1, N3-1) + P1\cdot TAB\ (N1-1, N2-1, N3-1)\right\}\Big]$$

Compact Input Sheet

N	Format	Variables	Card	No. of Columns
-1	35I1	KRD(I) I-1, 35		35
0	I3, 2X11A6, 1X11, L1, I4	I Run, HEADER, KA6, BODFIX, IDATE	Lead Card 1	78
1	6I2, F12.5	IMAX, JMAX, KMAX, LMAX, NRK, NCMAX, EPSMAX	Subscript Card	24
2	6F12.5	TABA(I)		12-72
3	↓	TABM(J)		↓
4	↓	TABAR(K)		↓
5	↓	CX (I, J) J then I		↓
6	↓	CN (I, J) J then I		↓
7	↓	CM (I, J) J then I		↓
8	↓	CMQ (I, J) J then I		↓
9	↓	CNR (I, J) J then I		↓
10	↓	CMPR (I, J) J then I		↓
11	↓	CMPQ (I, J) J then I		↓
12	↓	CNPA (I, J, K) K then J then I		↓
13	↓	CL (I, J, K) K then J then I		↓
14	↓	CLP (I, J, K) K then J then I		↓
15	↓	CMPA (I, J, K) K then J then I		↓
16	↓	G, DMM, DIX, DI, DIZ, DIXY	Lead Card 2	72
17	↓	S, DEE, THETA, PSI, Z, PHI	Lead Card 3	72
18	↓	XDOT, YDOT, ZDOT, P, Q, R	Lead Card 4	72
19	F12.5, 4XI8, 4F12.5, I4	TSTEP, INCPT1, TMAX, ZSTOP, TCH, TNEW, INCPT2	Lead Card 5	76
20	6F12.5	TABZ(L)		12-72
	↓	WDX(L)		12-72
	↓	WDY(L)		12-72
21	6F12.5	DY, ZETD1, ZETD2 ETA1, ETA2	Lead Card 6	60
22	↓	CYO (I, J) J then I		12-72
23	↓	CZO (I, J) J then I		↓
24	↓ Body	CMO (I, J) J then I		↓
25	↓ Axis	CNO (I, J) J then I		↓
26	↓ Only	CSF1 (I, J) J then I		↓
27	↓	CN1 (I, J) J then I		↓
28	↓	CSF3 (I, J) J then I		↓
29	↓	CSM1 (I, J) J then I		↓
30	↓	CM1 (I, J) J then I		↓
31	↓	CSM3 (I, J) J then I		↓
32	↓	CLPM1 (I, J) J then I		↓
33	↓	CLPM2 (I, J) J then I		↓

N is a code for card type (helpful to punch in columns 79-80)

N > 0 all optional on added runs (KRD(N) = 0 no, KRD(N) = 1 read)

N = 20 also optional on any run (LMAX = 0 no, LMAX > 0 read)

N = 0 required for each run.

N = -1 required for each additional run.

Cards 0 → 19 required for FIXED PLANE 1st run; 20 if LMAX > 0.

Cards 0 → 19, 21-33 required for BODY FIXED 1st run; 20 if LMAX > 0.

SECTION V

PROGRAM LISTING

```
$      FORTRAN NLSTOU,SYMTAB
$       INCODE   IBMF
C SIXDEG          SIXDEG (MAIN)                                                SIXD
C         MAIN PROGRAM FOR EXPANDED SIX-DEGREES-OF-FREEDOM TRAJECTORY         SIXD
C    INCLUDES  BODY AXIS ORIENTATION + EXPANDED AERO-DYNAMICS                 SIXD
C    INCLUDES  WIND EFFECTS                                                   SIXD
C                                                                             SIXD
COMMON BTOF,MPGIN,ALLIN,EQMPG                                                 COMMON
      COMMON/BTOF/ XMAT(3,3)          P,Q,R,ZALT,PHI,THETA,PSI,TSTEP,INCPT1COMMON
      COMMON /MPGIN/XDOT(3),                                                  COMMON
     *,TMAX,ZSTOP,TCH,TNEW,NRK,NCMAX,EPSMAX,ZETD1,ZETD2,INCPT2                COMMON
      COMMON /ALLIN/DIX,DI,DIZ,DIXY, BODFIX                                   COMMON
      COMMON /EQMPG/ QR,RAT,BODJX,BODJY,BODJZ,BODEN,BODJ,BODPD,BODQD,
     *BODRD,ZET1,ZET2                                                         SIXD
C                                                                             SIXD
      DIMENSION D(13,5),F(13,4),E(13),BD(13),DUM(4)                           SIXD
      LOGICAL QR,BODFIX                                                       SIXD
      CALL REPRT                                                              SIXD
    1 CALL READIN                                                             SIXD
      CERR=1./29.                                                             SIXD
      IF(NRK.EQ.3) CERR=1./7.                                                 SIXD
      IF(NRK.EQ.2) CERR=1./3.                                                 SIXD
      IF(NRK.LT.2 .OR.NRK.GT.4) NRK=4                                         SIXD
      DT=TSTEP                                                                SIXD
      INCPT=INCPT1                                                            SIXD
      QR=.TRUE.                                                               SIXD
      KOUNT=(TCH/DT)+.1                                                       SIXD
      KNT=0                                                                   SIXD
      LINE=100                                                                SIXD
      NSTO=5-NRK                                                              SIXD
      XINC=NRK-1                                                              SIXD
      TIME=0.                                                                 SIXD
      RAT=DIX/DI                                                              SIXD
      BD(4)=P                                                                 SIXD
      BD(5)=Q                                                                 SIXD
      BD(6)=R                                                                 SIXD
      BD(7)=0.                                                                SIXD
      BD(8)=0.
```

```
      BD(9)=ZALT                                                        SIXD
      SAVE=8.726646E-3*THETA                                            SIXD
      DUM(1)=COS(SAVE)                                                  SIXD
      DUM(2)=SIN(SAVE)                                                  SIXD
      SAVE=8.726646E-3*PSI                                              SIXD
      DUM(3)=COS(SAVE)                                                  SIXD
      DUM(4)=SIN(SAVE)                                                  SIXD
      BD(10)= DUM(1)*DUM(3)                                             SIXD
      BD(11)=-DUM(2)*DUM(4)                                             SIXD
      BD(12)= DUM(2)*DUM(3)                                             SIXD
      BD(13)= DUM(1)*DUM(4)                                             SIXD
      ZET1=0.                                                           SIXD
      ZET2=0.                                                           SIXD
      IF(.NOT.BODFIX) GO TO 10                                          SIXD
C    EXTRA CALCULATIONS FOR BODY-FIXED AXIS                             SIXD
      ZET1= 1.745329E-2*ZETD1                                           SIXD
      ZET2= 1.745329E-2*ZETD2                                           SIXD
      BODJX=DIXY/DIX                                                    SIXD
      BODJY=DIXY/DI                                                     SIXD
      BODJZ=DIXY/DIZ                                                    SIXD
      BODEN=1.-BODJX*BODJY                                              SIXD
      SAVE=(DIX+DI-DIZ)/DIX                                             SIXD
      BODJ=SAVE*BODJY                                                   SIXD
      BODPD=SAVE-BODEN                                                  SIXD
      BODQD=SAVE*RAT-BODEN                                              SIXD
      BODRD=(DIX-DI)/DIZ                                                SIXD
      KOUNT=-1                                                          SIXD
      IF(PHI.EQ.0.) GO TO 10                                            SIXD
      SAVE=8.726646E-3*PHI                                              SIXD
      DUM(1)=COS(SAVE)                                                  SIXD
      DUM(2)=SIN(SAVE)                                                  SIXD
      SAVE=BD(10)                                                       SIXD
      BD(10)=SAVE*DUM(1)-BD(11)*DUM(2)                                  SIXD
      BD(11)=SAVE*DUM(2)+BD(11)*DUM(1)                                  SIXD
      SAVE=BD(12)                                                       SIXD
      BD(12)=SAVE*DUM(1)+BD(13)*DUM(2)                                  SIXD
      BD(13)=-SAVE*DUM(2)+BD(13)*DUM(1)                                 SIXD
C                                                                       SIXD
   10 CALL SETMAX(BD(10))                                               SIXD
```

```
      IF(.NOT.BODFIX) XMAT(3,2)=0.                                      SIXD
C   GET U,V,W                                                           SIXD
      DO 14 I=1,3                                                       SIXD
      BD(I)=0.                                                          SIXD
      DO 14 J=1,3                                                       SIXD
   14 BD(I)=BD(I)+XMAT(J,I)*XDOT(J)                                     SIXD
      DO 19 I=1,13                                                      SIXD
      E(I)=0.                                                           SIXD
   19 D(I,NST0)=BD(I)                                                   SIXD
C   INITIAL START OR RE-START                                           SIXD
   50 NINC=MAX0(1,INCPT)                                                SIXD
      NST=NST0                                                          SIXD
      EPSIL=0.                                                          SIXD
      DTCOR=DT/3.                                                       SIXD
      DTTRP=2.*DT                                                       SIXD
      DTADM=DT/12.                                                      SIXD
      DTMIL=4.*DTCOR                                                    SIXD
  100 CALL EQMOT(D(1,NST),F(1,NST))                                     SIXD
C   STOP OR CHANGE EQ TESTS                                             SIXD
      IF(D(9,NST).GT.ZSTOP.OR.TIME.GE.TMAX) GO TO 102                   SIXD
      IF(NINC.LT.INCPT) GO TO 105                                       SIXD
C   TIME TO PRINT                                                       SIXD
      NINC=0                                                            SIXD
  102 CALL WRITIT(TIME,D(1,NST),E,F(1,NST),EPSIL,LINE)                  SIXD
      IF(NINC.GT.0) GO TO 1                                             SIXD
  105 IF(KNT.NE.KOUNT) GO TO 110                                        SIXD
CHANGE TO STEADY-STATE EQUATIONS FOR Q,R AND RAISE DT (OLD QRNOT)       SIXD
      WRITE(6,2105)
 2105 FORMAT(1X32(1H*),32HSIMPLIFIED EQUATIONS FOR Q AND R 32(1H*))
      KNT=KNT+1                                                         SIXD
      QR=.FALSE.                                                        SIXD
      IF(TNEW.GT.0.) DT=TNEW                                            SIXD
      IF(INCPT2.GT.0) INCPT=INCPT2                                      SIXD
      DO 109 I=1,13                                                     SIXD
      E(I)=0.                                                           SIXD
  109 D(I,NST0)=D(I,NST)                                                SIXD
      CALL EQMOT(D(1,NST0),F(1,NST0))                                   SIXD
      GO TO 50                                                          SIXD
C   ADVANCE TIME                                                        SIXD
```

```
  110 TIME=TIME+DT                                                      SIXD
      NINC=NINC+1                                                       SIXD
      KNT=KNT+1                                                         SIXD
      IF(NST.GE.4) GO TO 200                                            SIXD
CONTINUE BUILDING UP FIRST 2-4 FUNCTIONS                                SIXD
      CALL RUNGE(D(1,NST),F(1,NST),DT,D(1,NST+1))                       SIXD
      NST=NST+1                                                         SIXD
      IF(NST.LT.4.OR.NRK.EQ.2) GO TO 100                                SIXD
C   LAST RUNGE-KUTTA FUNCTION BUILT,CALCULATE FROM NSTO TO NST FOR ERRORSIXD
      CALL RUNGE(D(1,NSTO),F(1,NSTO),XINC*DT,D(1,5))                    SIXD
      GO TO 250                                                         SIXD
C   PREDICT                                                             SIXD
  200 DO 249 I=1,13                                                     SIXD
      GO TO (220,220,230,240),NRK                                       SIXD
C       TRAPAZOID                                                       SIXD
  220 D(I,5)=D(I,3)+DTTRP*F(I,4)                                        SIXD
      GO TO 245                                                         SIXD
C       ADAMS                                                           SIXD
  230 D(I,5)=D(I,4)+DTADM*(23.*F(I,4)-16.*F(I,3)+5.*F(I,2))             SIXD
      GO TO 245                                                         SIXD
C       MILNE                                                           SIXD
  240 D(I,5)=D(I,1)+DTMIL*(2.*(F(I,4)+F(I,2))-F(I,3))                   SIXD
  245 DO 247 J=1,3                                                      SIXD
      D(I,J)=D(I,J+1)                                                   SIXD
  247 F(I,J)=F(I,J+1)                                                   SIXD
  249 D(I,4)=D(I,5)                                                     SIXD
C   CORRECT WITH SIMPSON                                                SIXD
  250 NCOR=0                                                            SIXD
  260 CALL EQMOT(D(1,4),F(1,4))                                         SIXD
      DO 269 I=1,13                                                     SIXD
      IF(I.GE.10) DUM(I-9)=D(I,4)                                       SIXD
  269 D(I,4)=D(I,2)+DTCOR*(F(I,4)+4.*F(I,3)+F(I,2))                     SIXD
      CALL NORMAL(D(10,4))
      EPSIL=0.
      DO 274 I=10,13
  274 EPSIL=EPSIL+(D(I,4)-DUM(I-9))**2
      EPSIL=SQRT(EPSIL)
      NCOR=NCOR+1                                                       SIXD
CORRECT AGAIN IF ERROR IS TOO LARGE AND RE-CORRECTIONS ARE ALLOWED      SIXD
```

```
      IF(NCOR.LT.NCMAX.AND.EPSIL.GT.EPSMAX) GO TO 260                  SIXD
      DO 279 I=1,13
  279 E(I)=E(I)+CERR*(D(I,4)-D(I,5))                                   SIXD
      GO TO 100                                                        SIXD
      END                                                              SIXD
```

```
$      FORTRAN NLSTOU,SYMTAB
$       INCODE  IBMF
C REPRT          REPRT
      SUBROUTINE REPRT                                                  REPR0010
      DIMENSION DATA(16)                                                REPR0020
    1 FORMAT (16A5)                                                     REPR0030
    2 FORMAT (2H  ,I5,5X,16A5)                                          REPR0040
    3 FORMAT(20H1INPUT DATA --------/12H CARD NO.   ,1H1,7X,2H10,8X,2H20REPR0050
     1,8X,2H30,8X,2H40,8X,2H50,8X,2H60,8X,2H70,8X,2H80//)               REPR0060
C**************** KINDLY NOTE THAT COLS. 1-78 HAVE BEEN MOVED TO COLS. 3-80
C       IN  ORDER TO AVOID RE-READING PROBLEMS OFF OF UNIT 4 *******************
C  (THIS WILL CAUSE THE LAST 2 DIGOTS OF DATE TO DISAPPEAR IN READIN, NOTE THAT
C       THE ROUTINE WRITIT PLACES IN A 70 ON THE OUTPUT OF DATE) ****************
    4 FORMAT(2X,15A5,A3)
      N = 0                                                             REPR0070
      LINE = 56                                                         REPR0080
C************************************REPLACE BELOW**************************
      CALL FLGEOF(5,IEOF)                                               REPR0090
      GO TO 20                                                          REPR0100
C*****************************************WITH******************************
C     ASSIGN 10 TO IEOF               *****REMOVE C IN COL1*****************
C     IF(EOF(IEOF)) 10,20,20          *****REMOVE C IN COL1*****************
   10 END FILE 4                                                        REPR0110
      END FILE 4                                                        REPR0120
      REWIND 4                                                          REPR0130
      RETURN                                                            REPR0140
   20 REWIND 4                                                          REPR0150
   40 READ (5,1) (DATA(I),I=1,16)                                       REPR0160
C**********************************DELETE CARD BELOW***********************
      IF(IEOF.GT.0) GO TO 10                                            REPR0161
C**************************************************************************
      N = N + 1                                                         REPR0170
      IF(LINE-56) 50,30,30                                              REPR0180
   30 WRITE (6,3)                                                       REPR0190
      LINE = 3                                                          REPR0200
   50 WRITE (6,2) N , (DATA(I),I=1,16)                                  REPR0210
      WRITE (4,4) (DATA(I),I=1,16)                                      REPR0222
      LINE = LINE + 1                                                   REPR0230
      GO TO 40                                                          REPR0240
      END                                                               REPR0250
```

```
$      FORTRAN NLSTOW,SYMTAB
$       INCODE  IBMF
C READIN         READIN
      SUBROUTINE READIN                                                 READIN
C   SUBROUTINE TO READ IN DATA FOR ONE RUN AT A TIME                    READIN
CARD CODE TO PERMIT NEEDED DATA ONLY ON 2ND RUN.                        READIN
COMMON EQMIN,OUTIN,MPGIN,ALLIN                                          READIN
      COMMON /EQMIN/IMAX,JMAX,KMAX,LMAX,TABA(20),TABM(5),TABAR(5),TABZ(1COMMON
     *0),CX(20,5),CN(20,5),CM(20,5),CMQ(20,5),CNR(20,5),CMPR(20,5),CNPQ(COMMON
     *20,5),CNPA(20,5,5),CL(20,5,5),CLP(20,5,5),CMPA(20,5,5),           COMMON
     *CY0(20,5),CZ0(20,5),CM0(20,5),CN0(20,5),CSF1(20,5),CSF2(20,5),    COMMON
     *CSF3(20,5),CSM1(20,5),CSM2(20,5),CSM3(20,5),CLPH1(20,5),CLPH2(20,5COMMON
     *),WDX(10),WDY(10),G,DMM,S,DEE,DY,ETA1,ETA2                        COMMON
      COMMON /OUTIN/HEADER(11),IRUN,IDATE,KA6                           COMMON
      COMMON /MPGIN/XDCT(3),          P,Q,R,ZALT,PHI,THETA,PSI,TSTEP,INCPT1COMMON
     *,TMAX,ZSTOP,TCH,TNEW,NRK,NCMAX,EPSMAX,ZETD1,ZETD2,INCPT2          COMMON
      COMMON /ALLIN/DIX,DI,DIZ,DIXY, BODFIX                             COMMON
      DIMENSION KRD(35)                                                 READIN
      LOGICAL BODFIX                                                    READIN
      DATA NCARDS /0/                                                   READIN
C                                                                       READIN
C**************************************ADD BELOW*******************************
C     ASSIGN 9000 TO MEOF              *****REMOVE C IN COL1************************
C     IF(EOF(MEOF)) 9000,9000,7000 *****REMOVE C IN COL1************************
 7000 IF(NCARDS.GT.0) GO TO 7003                                        READIN
C   INITIAL READ                                                        READIN
C*************************************DELETE CARD BELOW**************************
      CALL FLGEOF(4,MEOF)                                               READIN
C*******************************************************************************
      NCARDS=33                                                         READIN
      DO 4 I=1,33                                                       READIN
    4 KRD(I)=1                                                          READIN
      GO TO 7002                                                        READIN
 7003 READ(4,208) (KRD(I),I=1,35)                                       READIN
C*************************************DELETE CARD BELOW**************************
      IF(MEOF.GT.0) GO TO 9000                                          READIN
C*******************************************************************************
 7002 READ(4,207)  IRUN,HEADER,KA6,BODFIX,IDATE                         READIN
```

```
      IF(KRD(1).GT.0) READ(4,201) IMAX,JMAX,KMAX,LMAX,NRK,NCMAX,EPSMAX   READIN
      IF(IMAX.GT.20.OR.JMAX.GT.5.OR.KMAX.GT.5.OR.LMAX.GT.10) GO TO 8001 READIN
      IF(KRD(2).GT.0) READ(4,202) (TABA(I),I=1,IMAX)
      IF(KRD(3).GT.0) READ(4,202) (TABM(I),I=1,JMAX)
      IF(KRD(4).GT.0) READ(4,202) (TABAR(I),I=1,KMAX)
      IF(KRD(5).GT.0) READ(4,202) ((CX(I,J),J=1,JMAX),I=1,IMAX)
      IF(KRD(6).GT.0) READ(4,202) ((CN(I,J),J=1,JMAX),I=1,IMAX)
      IF(KRD(7).GT.0) READ(4,202) ((CM(I,J),J=1,JMAX),I=1,IMAX)
      IF(KRD(8).GT.0) READ(4,202) ((CMQ(I,J),J=1,JMAX),I=1,IMAX)
      IF(KRD(9).GT.0) READ(4,202) ((CNR(I,J),J=1,JMAX),I=1,IMAX)
      IF(KRD(10).GT.0) READ(4,202) ((CMPR(I,J),J=1,JMAX),I=1,IMAX)
      IF(KRD(11).GT.0) READ(4,202) ((CNPQ(I,J),J=1,JMAX),I=1,IMAX)
      IF(KRD(12).GT.0) READ(4,202) (((CNPA(I,J,K),K=1,KMAX),J=1,JMAX),
     1I=1,IMAX)
      IF(KRD(13).GT.0) READ(4,202) (((CL(I,J,K),K=1,KMAX),J=1,JMAX),
     1I=1,IMAX)
      IF(KRD(14).GT.0) READ(4,202) (((CLP(I,J,K),K=1,KMAX),J=1,JMAX),
     1I=1,IMAX)
      IF(KRD(15).GT.0) READ(4,202) (((CMPA(I,J,K),K=1,KMAX),J=1,JMAX),
     1I=1,IMAX)
      IF(KRD(16).GT.0) READ(4,202) G,DMM,DIX,DI,DIZ,DIXY                READIN
      IF(KRD(17).GT.0) READ(4,202) S,DEE,THETA,PSI,ZALT,PHI             READIN
      IF(KRD(18).GT.0) READ(4,202)(XDOT(I),I=1,3),P,Q,R                 READIN
      IF(KRD(19).GT.0)READ(4,206)TSTEP,INCPT1,TMAX,ZSTOP,TCH,TNEW,INCPT2
      IF(KRD(20).EQ.0.OR.LMAX.EQ.0) GO TO 10                            READIN
      READ(4,202) (TABZ(I),I=1,LMAX)                                    READIN
      READ(4,202) (WDX(I),I=1,LMAX)                                     READIN
      READ(4,202) (WDY(I),I=1,LMAX)                                     READIN
   10 IF(.NOT.BODFIX) RETURN                                            READIN
      IF(KRD(21).GT.0) READ(4,202) DY,ZETD1,ZETD2,ETA1,ETA2             READIN
      IF(KRD(22).GT.0) READ(4,202) ((  CY0(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(23).GT.0) READ(4,202) ((  CZ0(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(24).GT.0) READ(4,202) ((  CM0(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(25).GT.0) READ(4,202) ((  CN0(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(26).GT.0) READ(4,202) (( CSF1(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(27).GT.0) READ(4,202) (( CSF2(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(28).GT.0) READ(4,202) (( CSF3(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(29).GT.0) READ(4,202) (( CSM1(I,J),J=1,JMAX),I=1,IMAX)     READIN
      IF(KRD(30).GT.0) READ(4,202) (( CSM2(I,J),J=1,JMAX),I=1,IMAX)     READIN
```

```
      IF(KRD(31).GT.0) READ(4,202) (( CSM3(I,J),J=1,JMAX),I=1,IMAX)    READIN
      IF(KRD(32).GT.0) READ(4,202) ((CLPH1(I,J),J=1,JMAX),I=1,IMAX)    READIN
      IF(KRD(33).GT.0) READ(4,202) ((CLPH2(I,J),J=1,JMAX),I=1,IMAX)    READIN
      RETURN                                                           READIN
 8001 WRITE(6,8011)                                                    READIN
 9000 CALL EXIT                                                        READIN
  208 FORMAT(2X,35I1)
  207 FORMAT(2X,I3,2X,11A6,1X,I1,L1,A4)                                READIN
  201 FORMAT(2X,6I2,F12.5)
  202 FORMAT(2X,6F12.5)
  206 FORMAT(2X,F12.5,4X,I8,4F12.5,I4)
 8011 FORMAT(1H ,26H SUBSCRIPT VALUE TOO LARGE)                        READIN
      END                                                              READIN
```

```
$      FORTRAN NLSTOU,SYMTAB
$       INCODE  IBMF
C RUNGE          RUNGE
      SUBROUTINE RUNGE(D,F,DT,G)                                        RUNGE
C     SUBROUTINE TO NUMERICALLY INTEGRATE VIA RUNGE-KUTTA (3RD ORDER)   RUNGE
      DIMENSION D(13),F(13),G(13),F2(13),F3(13)                         RUNGE
      H=DT                                                              RUNGE
      HBY2=H/2.                                                         RUNGE
      HBY6=H/6.                                                         RUNGE
      DO 3 I=1,13                                                       RUNGE
    3 G(I)=D(I)+HBY2*F(I)                                               RUNGE
      CALL EQMOT(G,F2)                                                  RUNGE
      DO 5 I=1,13                                                       RUNGE
    5 G(I)=D(I)+H*(2.*F2(I)-F(I))                                       RUNGE
      CALL EQMOT(G,F3)                                                  RUNGE
      DO 7 I=1,13                                                       RUNGE
    7 G(I)=D(I)+HBY6*(F(I)+4.*F2(I)+F3(I))                              RUNGE
      RETURN                                                            RUNGE
      END                                                               RUNGE
```

```
$       FORTRAN NLSTOU,SYNTAE
$        INCODE  IBMF
C EQMOT           EQMOT
      SUBROUTINE EQMOT(Y,Z)                                           EQMOT
C   SUBROUTINE TO DETERMINE THE DERIVATIVES OF THE 13 BASIC VARIABLES  EQMOT
COMMON TBL,BTOF,EQMIN,ALLIN,EQMPG,OUTEM                               EQMOT
      COMMON/TBL/ N1,N2,N3,P1,P2,P3                                   COMMON
      COMMON/BTOF/ XMAT(3,3)                                          COMMON
      COMMON /EQMIN/IMAX,JMAX,KMAX,LMAX,TABA(20),TABM(5),TABAR(5),TABZ(1COMMON
     *0),CX(20,5),CN(20,5),CN(20,5),CMQ(20,5),CNR(20,5),CMPR(20,5),CNPQ(COMMON
     *20,5),CNFA(20,5,5),CL(20,5,5),CLP(20,5,5),CMPA(20,5,5),         COMMON
     *CY0(20,5),CZ0(20,5),CM0(20,5),CN0(20,5),CSF1(20,5),CSF2(20,5),   COMMON
     *CSF3(20,5),CSM1(20,5),CSM2(20,5),CSM3(20,5),CLPH1(20,5),CLPH2(20,5COMMON
     *),WDX(10),WDY(10),G,DMM,S,DEE,DY,ETA1,ETA2                      COMMON
      COMMON /ALLIN/DIX,DI,DIZ,DIXY, BODFIX                           COMMON
      COMMON /EQMPG/ QR,PAT,BODJX,BODJY,BODJZ,BODEN,BODJ,BODPD,BODQD,  COMMON
     *BODRD,ZET1,ZET2
      COMMON /OUTEM/ CAPHI1,CAPHI2,PDV,ALPHA,CAPVA,EM,PFE             COMMON
      DIMENSION Y(13),Z(13)                                           EQMOT
      DIMENSION VA(3),XDOTA(3)                                        EQMOT
      LOGICAL GR,BODFIX                                               EQMOT
      CALL NORMAL(Y(10))                                              EQMOT
      CALL SETMAX(Y(10))                                              EQMOT
      IF(.NOT.BODFIX) XMAT(3,2)=0.                                    EQMOT
      DO 3 I=1,3                                                      EQMOT
      Z(I+6)=0.                                                       EQMOT
      DO 3 J=1,3                                                      EQMOT
    3 Z(I+6)=Z(I+6)+XMAT(I,J)*Y(J)                                    EQMOT
      IF(LMAX.GT.0) GO TO 6                                           EQMOT
      DO 5 I=1,3                                                      EQMOT
    5 VA(I)=Y(I)                                                      EQMOT
      GO TO 10                                                        EQMOT
    6 CALL GETIN(Y(9),TABZ,LMAX,N1,P1)                                EQMOT
      XDOTA(1)=Z(7)-TBL1(WDX)                                         EQMOT
      XDOTA(2)=Z(8)-TBL1(WDY)                                         EQMOT
      XDOTA(3)=Z(9)                                                   EQMOT
      DO 7 I=1,3                                                      EQMOT
      VA(I)=0.                                                        EQMOT
```

```
      DO 7 J=1,3                                                         EQMOT
    7 VA(I)=VA(I)+XMAT(J,I)*XDOTA(J)                                     EQMOT
   10 VASQ=VA(2)**2+VA(3)**2                                             EQMOT
      UASQ=VA(1)**2+VASQ                                                 EQMOT
      ALPHA=0.                                                           EQMOT
      IF(UASQ.GT.1.E-8) GO TO 12                                         EQMOT
C    VELOCITY=0,SET OUTPUT TO ZERO AND SKIP AERODYNAMICS                 EQMOT
      CAPVA=0.                                                           EQMOT
      PDV=-999999.0                                                      EQMOT
      CAPHI1=-ZET1                                                       EQMOT
      CAPHI2=-ZET2                                                       EQMOT
      ZETA=0.                                                            EQMOT
      EM=0.                                                              EQMOT
      FX=0.                                                              EQMOT
      FY=0.                                                              EQMOT
      FZ=0.                                                              EQMOT
   11 EMX=0.                                                             EQMOT
      EMY=0.                                                             EQMOT
      EMZ=0.                                                             EQMOT
      IF(CR) GO TO 50                                                    EQMOT
      Y(5)=0.                                                            EQMOT
      Y(6)=0.                                                            EQMOT
      GO TO 48                                                           EQMOT
   12 CAPVA=SQRT(UASQ)                                                   EQMOT
      IF(VASQ.GT.1.E-8) GO TO 14                                         EQMOT
C    SIDE VEL=0,SET FUNCTIONS OF ZETA TO 0.                              EQMOT
      ZETA=0.                                                            EQMOT
      SINZE=0.                                                           EQMOT
      COSZE=0.                                                           EQMOT
      GO TO 20                                                           EQMOT
   14 DEN=SQRT(VASQ)                                                     EQMOT
      ALPHA=ATAN2(DEN,VA(1))                                             EQMOT
      SINZE=VA(3)/DEN                                                    EQMOT
      COSZE=VA(2)/DEN                                                    EQMOT
      ZETA=ATAN2(SINZE,COSZE)                                            EQMOT
   20 IF(36000.+Y(9).LE.0.) GO TO 22                                     EQMOT
C LOW ALTITUDE                                                           EQMOT
      VOS=1116.89+0.004123*Y(9)                                          EQMOT
      RHO=0.0023769*(1.+6.875E-6*Y(9))**4.2561                           EQMOT
```

```
      GO TO 25                                                          EQMOT
C HIGH ALTITUDE                                                         EQMOT
   22 VOS=968.46                                                        EQMOT
      RHO=0.004*EXP(48..6E-6*Y(9))                                      EQMOT
   25 EM=CAPVA/VOS                                                      EQMOT
C     SET-UP TABLE VALUES                                               EQMOT
      PDV=.5*Y(4)*DEE/CAPVA                                             EQMOT
      CALL GETIN(ALPHA,TAPA,IMAX,N1,P1)                                 EQMOT
      CALL GETIN(EM    ,TARM,JMAX,N2,P2)                                EQMOT
      CALL GETIN(PDV  ,TABAR,KMAX,N3,P3)                                EQMOT
      CON=.5*RHO*UASQ*S                                                 EQMOT
      FX=CON*TBL2(CX)                                                   EQMOT
      CCN=TBL2(CN)                                                      EQMOT
      CNPADD=-TBL3(CNPA)*PDV                                            EQMOT
      CAPHI1=1.5707963-ZETA+ZET1                                        EQMOT
      CAPHI2=1.5707963-ZETA+ZET2                                        EQMOT
      IF(.NOT.BODFIX) GO TO 30                                          EQMOT
      SET1=SIN(ETA1*CAPHI1)                                             EQMOT
      SET2=SIN(ETA2*CAPHI2)                                             EQMOT
      CNPADD=-CNPADD+TBL2(CSF1)*SET1                     +TBL2(CSF3)    EQMOT
      CCN=                                CCN+TBL2(CSF2)*SET1           EQMOT
   30 FY=-CCN    SZE+CNPADD*SINZE                                       EQMOT
      IF(BODF     FY=FY+TBL2(CY0)                                       EQMOT
      FY=CON*FY                                                         EQMOT
      FZ=-CCN*SINZE-CNPADD*COSZE                                        EQMOT
      IF(BODFIX) FZ=FZ+TBL2(CZ0)                                        EQMOT
      FZ=CON*FZ                                                         EQMOT
      CON=CON*DEE                                                       EQMOT
      EMX=TBL3(CL)+PDV*TBL3(CLP)                                        EQMOT
      IF(BODFIX) EMX=EMX+TBL2(CLPH1)*SET1+TBL2(CLPH2)*SET2              EQMOT
      EMX=EMX*CON                                                       EQMOT
      IF(BODFIX.AND.DY.NE.0.) EMX=EMX-DY*FZ                             EQMOT
      EMX=EMX/DIX                                                       EQMOT
      SAVE2=.5*DEE/CAPVA                                                EQMOT
      SAVE1=(Y(5)*SINZE-Y(6)*COSZE)*SAVE2                               EQMOT
      SAVE2=(Y(6)*SINZE+Y(5)*COSZE)*SAVE2                               EQMOT
      CMADD=TBL2(CM)+SAVE1*TBL2(CMQ)+SAVE2*TBL2(CMPR)*PDV               EQMOT
      CMPADD=SAVE2*TBL2(CNR)+(TBL3(CMPA)+SAVE1*TBL2(CNPQ))*PDV          EQMOT
      IF(BODFIX) CMPADD=CMPADD+TBL2(CSM1)*SET1+TBL2(CSM3)               EQMOT
```

```
      IF(BODFIX) CMADD=CMADD+TBL2(CSM2)*SET1                            EQMOT
      EMY=CMADD*SINZE+CMPADD*COSZE                                      EQMOT
      IF(BODFIX) EMY=EMY+TBL2(CM0)                                      EQMOT
      EMY=CON*EMY/DI                                                    EQMOT
      EMZ=-CMADD*COSZE+CMPADD*SINZE                                     EQMOT
      IF(BODFIX) EMZ=EMZ+TBL2(CN0)                                      EQMOT
      EMZ=EMZ*CON                                                       EQMOT
      IF(.NOT.BODFIX) EMZ=EMZ/DI                                        EQMOT
      IF(BODFIX) EMZ=(EMZ+DY*FX)/DIZ                                    EQMOT
   45 IF(QR) GO TO 50                                                   EQMOT
C    SPECIAL CALCULATION FOR Q AND R                                    EQMOT
      SAVE1=XMAT(3,1)/XMAT(3,3)*2.                                      EQMOT
      Y(6)=((RAT*Y(4))-SQRT(AMAX1(0.,(RAT*Y(4))**2-2.*SAVE1*EMY)))/SAVE1EQMOT
      Y(5)=-Y(6)*EMZ/EMY                                                EQMOT
   48 Z(5)=0.                                                           EQMOT
      Z(6)=0.                                                           EQMOT
CALCULATE EQUATIONS OF MOTION  FOR U,V,W,P,Q,R AND THE QUATERNIANS      EQMOT
   50 PEE=Y(4)                                                          EQMOT
      IF(.NOT.BODFIX) PEE=XMAT(3,1)/XMAT(3,3)*Y(6)                      EQMOT
      Z(1)=Y(2)*Y(6)-Y(3)*Y(5)+G*XMAT(3,1)+FX/DMM                       EQMOT
      Z(2)=Y(3)*PEE -Y(1)*Y(6)+G*XMAT(3,2)+FY/DMM                       EQMOT
      Z(3)=Y(1)*Y(5)-Y(2)*PEE +G*XMAT(3,3)+FZ/DMM                       EQMOT
      IF(BODFIX) GO TO 55                                               EQMOT
      Z(4)=EMX                                                          EQMOT
      SAVE1=PEE-DIX/DI*Y(4)                                             EQMOT
      Z(5)=EMY+SAVE1*Y(6)                                               EQMOT
      Z(6)=EMZ-SAVE1*Y(5)                                               EQMOT
      GO TO 60                                                          EQMOT
   55 Z(4)=((BODPD*Y(5)- BODJ*PEE)*Y(6)+EMX+BODJX*EMY)/BODEN            EQMOT
      Z(5)=(( BODJ*Y(5)-BODQD*PEE)*Y(6)+EMY+BODJY*EMX)/BODEN            EQMOT
      Z(6)=BODJZ*(PEE**2-Y(5)**2)+BODRD*PEE*Y(5)+EMZ                    EQMOT
   60 Z(10)=.5*(-Y(11)*PEE-Y(12)*Y(5)-Y(13)*Y(6))                       EQMOT
      Z(11)=.5*( Y(10)*PEE-Y(13)*Y(5)+Y(12)*Y(6))                       EQMOT
      Z(12)=.5*( Y(13)*PEE+Y(10)*Y(5)-Y(11)*Y(6))                       EQMOT
      Z(13)=.5*(-Y(12)*PEE+Y(11)*Y(5)+Y(10)*Y(6))                       EQMOT
      RETURN                                                            EQMOT
      END                                                               EQMOT
```

```
$       FORTRAN NLSTOU,SYMTAP
$        INCODE  IBMF
C WRITIT         WRITIT
      SUBROUTINE WRITIT(TIME,Y,E,Z,EPSIL,LINE)                          WRITIT
C   SUBROUTINE TO PRINT-OUT DESIRED DATA                                WRITIT
COMMON BTOF,OUTIN,OUTEM                                                 WRITIT
      COMMON/BTOF/ XMAT(3,3)                                            COMMON
      COMMON /OUTIN/HEADER(11),IRUN,IDATE,KA6                           COMMON
      COMMON /OUTEM/ CAPHI1,CAPHI2,PDV,ALPHA,CAPVA,EM,PEE               COMMON
      DIMENSION Y(13),E(13),Z(13)                                       WRITIT
C                                                                       WRITIT
      CAPV=SQRT(Y(1)**2+Y(2)**2+Y(3)**2)                                WRITIT
      ALD= 57.29578*ALPHA                                               WRITIT
      CP1= 57.29578*CAPHI1                                              WRITIT
      CP2= 57.29578*CAPHI2                                              WRITIT
      PHD= 57.29578*ATAN2(XMAT(3,2),XMAT(3,3))                          WRITIT
      THD= 57.29578*ATAN2(-XMAT(3,1),SQRT(XMAT(3,2)**2+XMAT(3,3)**2))   WRITIT
      PSD= 57.29578*ATAN2(XMAT(2,1),XMAT(1,1))                          WRITIT
      IF(LINE.LT.56) GO TO 5                                            WRITIT
      LINE=8                                                            WRITIT
      WRITE(6,100) HEADER,IRUN,IDATE                                    WRITIT
      IF(KA6.GT.0) WRITE(26,103) HEADER,IRUN,IDATE                      WRITIT
    5 LINE=LINE+2                                                       WRITIT
      WRITE(6,101) TIME,CAPV,(Y(I),I=7,9),(Z(I),I=7,9),ALD,CP1,CP2,CAPVAWRITIT
     *,EM,  (Y(I),I=1,6),PHD,THD,PSD,EPSIL                              WRITIT
      IF(KA6.LE.0) RETURN                                               WRITIT
      SAVE=1.-XMAT(3,1)**2                                              WRITIT
      IF(SAVE.GT.1.E-8) GO TO 10                                        WRITIT
      PHD=-9999999.9                                                    WRITIT
      THD=-9999999.9                                                    WRITIT
      PSD=-9999999.9                                                    WRITIT
      GO TO 20                                                          WRITIT
   10 PSD=(XMAT(3,2)*Y(5)+XMAT(3,3)*Y(6))/SAVE                          WRITIT
      SAVE=SQRT(SAVE)                                                   WRITIT
      THD=(XMAT(3,3)*Y(5)-XMAT(3,2)*Y(6))/SAVE                          WRITIT
      PHD=PEE-XMAT(3,1)*PSD                                             WRITIT
   20 WRITE(26,104) TIME,PHD,THD,PSD,(Y(I),I=10,13),(E(I),I=4,6),PDV,   WRITIT
     * (E(I),I=10,13)                                                   WRITIT
                                                                        WRITIT
```

```
      RETURN                                                            WRITIT
  100 FORMAT(1H1,31X,11A6/10X,11HRUN NUMBER ,I3,90X,5HDATE-,A4,3H/70 /  WRITIT
     * 4X4HTIME,
     16X8HVELOCITY,8X1HX,11X1HY,11X1HZ,9X4HXDOT,8X4HYDOT,9X4HZDOT,1X    WRITIT
     *21H  ALPHA   CPHI1   CPHI27X8HVA   MACH                           WRITIT
     2   /4X5H(SEC),5X8H(FT/SEC),6X4H(FT),8X4H(FT),8X4H(FT),6X8H(FT/SEC)WRITIT
     32(4X8H(FT/SEC)),3(7H   (DEG)),3X8H(FT/SEC)/12X1HU11X1HV11X1HW11X1HPWRITIT
     411X1HQ,11X1HR,9X3HPHI8X                                           WRITIT
     *                   5HTHETA,7X3HPSI,8X5HEPSIL/9X8H(FT/SEC),4X8H(FT/SECWRITIT
     5),4X8H(FT/SEC),3X9H(RAD/SEC),3X9H(RAD/SEC),3X9H(RAD/SEC),5X5H(DEG)WRITIT
     6,6X5H(DEG)6X5H(DEG) //)                                           WRITIT
  101 FORMAT(F9.4,7F12.1,1X      3F7.2,F10.1,F7.2/3X3F12.1,1X3F12.2,     WRITIT
     1 3F11.2,1PE17.6)                                                  WRITIT
  103 FORMAT(1H1,31X,11A6/10X,11HRUN NUMBER ,I3,90X,5HDATE-,A4,3H/70 /  WRITIT
     14X4HTIME12X5HPHDOT10X5HTHDOT10X5HPSDOT9X21(1H*),                  WRITIT
     *17HQUATERNION VALUES22(1H*)/4X5H(SEC)3X3(6X9H(RAD/SEC)),14X1H114X WRITIT
     *1H214X1H314X1H4/12X5HP-ERR10X5HQ-ERR10X5HR-ERR10X5HPD/2V2X21(1H*),WRITIT
     *17HQUATERNION ERRORS22(1H*)/4X3(6X9H(RAD/SEC)),22X1H114X1H214X1H3 WRITIT
     *14X1H4//)                                                         WRITIT
  104 FORMAT(F9.4,    3F15.2,4X4F15.6/4X1P8E15.6)                       WRITIT
      END                                                               WRITIT
```

```
$       FORTRAN NLSTOU,SYMTAB
$       INCODE  IBMF
C SETMAX        SETMAX
      SUBROUTINE SETMAY(XL)                                                 SETMAX
C   SUBROUTINE TO SET LP MATRIX FOR ROTATING BODY AXIS TO FIXED SPACE       SETMAX
C       XL=ARRAY OF QUATERNIANS  (L0,L1,L2,L3) =L-BAR                       SETMAX
C       XMAT=3X3 MATRIX---A FUNCTION OF L-BAR OR PHI,THETA,PSI              SETMAX
COMMON BTOF                                                                 SETMAX
      COMMON/BTOF/ XMAT(3,3)                                                COMMON
      DIMENSION XL(4)                                                       SETMAX
      XMAT(1,2)=XL(1)**2                                                    SETMAX
      XMAT(1,3)=XL(2)**2                                                    SETMAX
      SAVE1    =XL(3)**2                                                    SETMAX
      SAVE2    =XL(4)**2                                                    SETMAX
      XMAT(1,1)=XMAT(1,2)+XMAT(1,3)-SAVE1-SAVE2                             SETMAX
      SAVE2=SAVE1-SAVE2                                                     SETMAX
      SAVE1=XMAT(1,2)-XMAT(1,3)                                             SETMAX
      XMAT(2,2)=SAVE1+SAVE2                                                 SETMAX
      XMAT(3,3)=SAVE1-SAVE2                                                 SETMAX
      SAVE1=2.*XL(2)*XL(3)                                                  SETMAX
      SAVE2=2.*XL(1)*XL(4)                                                  SETMAX
      XMAT(1,2)=SAVE1-SAVE2                                                 SETMAX
      XMAT(2,1)=SAVE1+SAVE2                                                 SETMAX
      SAVE1=2.*XL(2)*XL(4)                                                  SETMAX
      SAVE2=2.*XL(1)*XL(3)                                                  SETMAX
      XMAT(1,3)=SAVE1+SAVE2                                                 SETMAX
      XMAT(3,1)=SAVE1-SAVE2                                                 SETMAX
      SAVE1=2.*XL(3)*XL(4)                                                  SETMAX
      SAVE2=2.*XL(1)*XL(2)                                                  SETMAX
      XMAT(2,3)=SAVE1-SAVE2                                                 SETMAX
      XMAT(3,2)=SAVE1+SAVE2                                                 SETMAX
      RETURN                                                                SETMAX
      END                                                                   SETMAX
```

```
$       FORTRAN NLSTOU,SYMTAB
$       INCODE  IBMF
C NORMAL        NORMAL
      SUBROUTINE NORMAL(XL)                                          NORMAL
C     SUBROUTINE TO NORMALIZE LAMBDAS SO THEIR DOT PRODUCT=1         NORMAL
      DIMENSION XL(4)                                                NORMAL
      SAVE=0.                                                        NORMAL
      DO 3 I=1,4                                                     NORMAL
    3 SAVE=SAVE+XL(I)**2                                             NORMAL
      SAVE=1./SQRT(SAVE)                                             NORMAL
      DO 5 I=1,4                                                     NORMAL
    5 XL(I)=XL(I)*SAVE                                               NORMAL
      RETURN                                                         NORMAL
      END                                                            NORMAL
```

```
$      FORTRAN NLSTOU,SYMTAB
$       INCODE  IBMF
C GETIN          GETIN
      SUBROUTINE GETIN(X,XTAB,NMAX,N,P)                                 GETIN
C    SUBROUTINE TO GET INDEX N AND DISTANCE P FROM N,SO THAT N IS THE 1STGETIN
C     INDEX WHERE X IS LT XTAB(N). TABLE IS SYMETRIC ABOUT XTAB(1)      GETIN
      DIMENSION XTAB(2)                                                 GETIN
      N=1                                                               GETIN
      IF(NMAX.LE.1) GO TO 6                                             GETIN
      ARG=X                                                             GETIN
      IF(ARG.LT.XTAB(1).AND.XTAB(1).GE.0.) ARG=-ARG                     GETIN
      IF(ARG.LE.XTAB(1)) GO TO 6                                        GETIN
      DO 4 N=2,NMAX                                                     GETIN
      SAVE=XTAB(N)-ARG                                                  GETIN
      IF(SAVE) 4,6,10                                                   GETIN
    4 CONTINUE                                                          GETIN
      N=NMAX                                                            GETIN
    6 P=0.                                                              GETIN
      RETURN                                                            GETIN
   10 P=SAVE/(XTAB(N)-XTAB(N-1) )                                       GETIN
      RETURN                                                            GETIN
      END                                                               GETIN
```

```
$       FORTRAN NLSTOU,SYMTAB
$        INCODE   IBMF
C TBL1               FUNCTION TAB1
      FUNCTION TBL1(TAB)                                                TBL1
C    SINGLE DIMENSIONED INTERPOLATION GIVEN N1 AND P1                   TBL1
COMMON TBL                                                              TBL1
      COMMON/TBL/ N1,N2,N3,P1,P2,P3                                     COMMON
      DIMENSION TAB(2)                                                  TBL1
      TBL1=TAB(N1)                                                      TBL1
      IF(P1.EQ.0.) RETURN                                               TBL1
      TBL1=TBL1-P1*(TBL1-TAB(N1-1))                                     TBL1
      RETURN                                                            TBL1
      END                                                               TBL1
```

```
$       FORTRAN NLSTOU,SYMTAB
$       INCODE  IBMF
C TBL2          FUNCTION TAB2
      FUNCTION TBL2(TAB)                                                TBL2
C   DOUBLE DIMENSIONED INTERPOLATION GIVEN N1,N2, AND P1,P2             TBL2
COMMON TBL                                                              TBL2
      COMMON/TBL/ N1,N2,N3,P1,P2,P3                                     COMMON
      DIMENSION TAB(20,5)                                               TBL2
      TBL2=TBL1(TAB(1,N2))                                              TBL2
      IF(P2.EQ.0.) RETURN                                               TBL2
      TBL2=TBL2-P2*(TBL2-TBL1(TAB(1,N2-1)))                             TBL2
      RETURN                                                            TBL2
      END                                                               TBL2
```

```
$       FORTRAN NLSTOU,SYMTAB
$       INCODE  IBMF
C TBL3           FUNCTION TAB3
      FUNCTION TBL3(TAB)
C   TRIPLE DIMENSIONED INTERPOLATION GIVEN N1,N2,N3 AND P1,P2,P3      TBL3
COMMON TBL                                                           TBL3
      COMMON/TBL/ N1,N2,N3,P1,P2,P3                                  TBL3
      DIMENSION TAB(20,5,5)                                          COMMON
      TBL3=TBL2(TAB(1,1,N3))                                         TBL3
      IF(P3.EQ.0.) RETURN                                            TBL3
      TBL3=TBL3-P3*(TBL3-TBL2(TAB(1,1,N3-1)))                        TBL3
      RETURN                                                         TBL3
      END                                                            TBL3
```


(The reverse of this page is blank)

SECTION VI

COMMENTS AND SPECIAL INSTRUCTIONS

A. MAGNUS ROTOR TRAJECTORY AND MOTION SIMULATION

In most instances, magnus rotor trajectories will be computed with fixed-plane axes. Where complete trajectory data are to be obtained (launch to impact) for flight times of several seconds or more, it will often be convenient to use the $\dot{q} = \dot{r} = 0$ option after about the first second of flight* (or after a time where the nutation is damped to a few degrees amplitude). For the investigation of autorotation initiation and the effects of dynamic unbalance, the body-fixed axes may be used.

For initiating magnus rotor trajectories, it is convenient to use the initial horizontal velocity component along the (-Y) axis, such that the Euler angles, θ and ψ , will approach zero if the rotor is in gliding flight attitude, $\tilde{\alpha} \rightarrow \pi/2$. In this manner, positive spin rates, corresponding to positive values of the spin torque coefficient $C_{\ell}(\tilde{\alpha}, M, \frac{pd}{2V})$, will result in an upward magnus lift force.

Special care must be used in selecting the initial conditions. Letting the initial flight path angle, γ , be specified by

$$\begin{aligned} \dot{X} &= 0 \\ -\dot{Y} &= V \cos \gamma \\ \dot{Z} &= V \sin \gamma \end{aligned} \qquad 1)$$

then the initial Euler angles, θ and ψ for the fixed-plane axes, are related to the initial sideslip angle, β , and the orientation of the angle of attack plane, $\bar{\varphi}$, by the following relations (see also Figure 6).

$$\sin(-\theta) = \cos\beta \sin\bar{\varphi} \cos\gamma + \sin\beta \sin\gamma \qquad 2)$$

$$\tan(-\psi) = \frac{-\cos\beta \sin\bar{\varphi} \sin\gamma + \sin\beta \cos\gamma}{\cos\beta \cos\bar{\varphi}} \qquad 3)$$

where $-\frac{\pi}{2} < \psi < \frac{\pi}{2}$ for $-\frac{\pi}{2} < \bar{\varphi} < \frac{\pi}{2}$

and $\frac{\pi}{2} < \psi < \frac{\pi}{2}$ for $\frac{\pi}{2} < \bar{\varphi} < \frac{3\pi}{2}$

---

* Note: if the $\dot{q} = \dot{r} = 0$ option is not used with fixed plane axes, then TCH > TSTOP for input lead card 5.

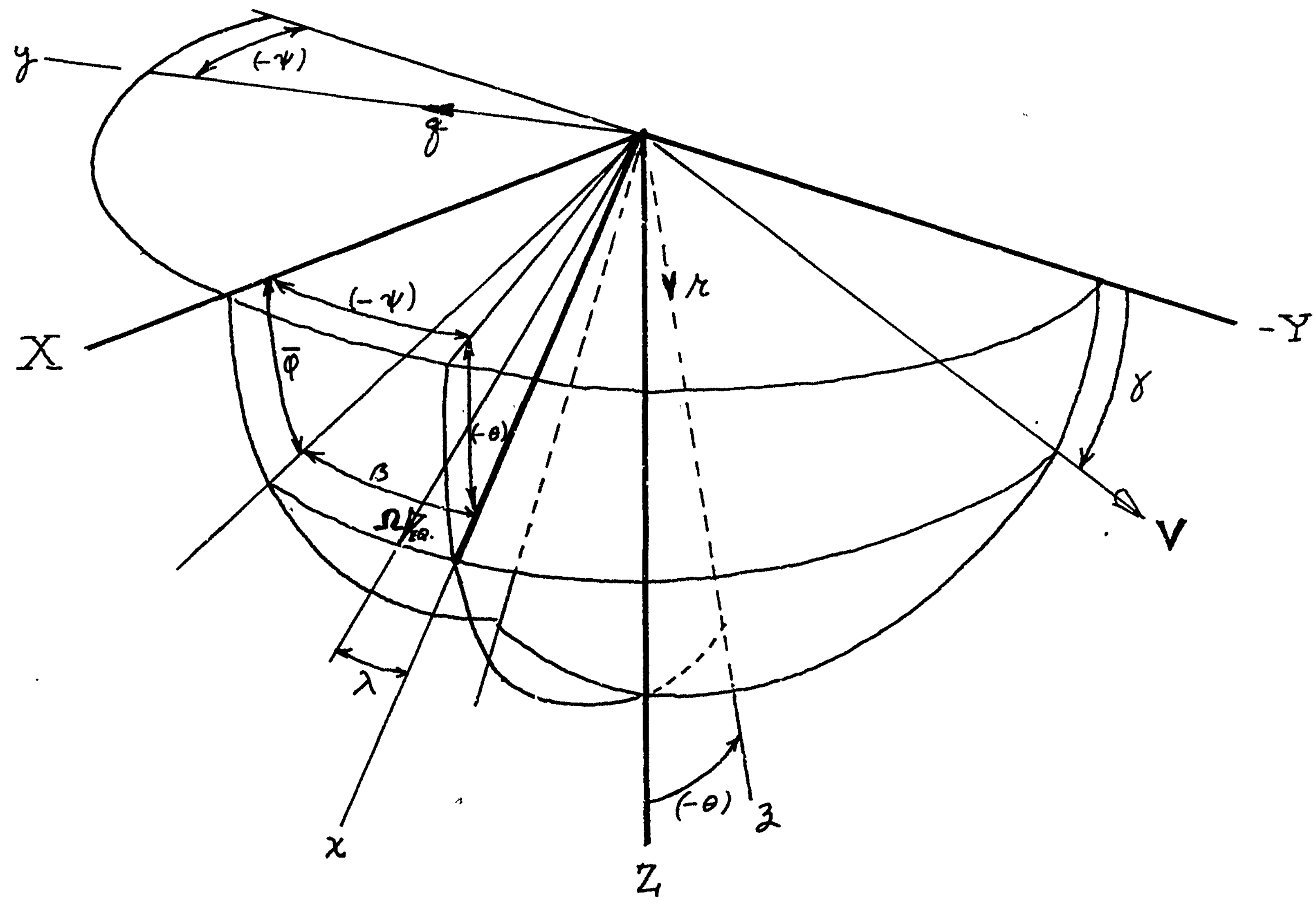


Figure 6. Description of Initial Conditions as a Function of γ , β , $\bar{\varphi}$ and λ

If an initial coning motion is desired, such that the total angular momentum does not coincide with x axis, then the coning angle can be specified by, λ , and the equivalent cross-angular velocity determined by

$$\Omega_{EQUIV.} = \frac{p\,\frac{I_x}{I}}{\cot\lambda} \qquad 4)$$

The angular velocity components with respect to the lateral fixed-plane axes are given in terms of $\Omega_{eq.}$ by the relationships

$$q = \Omega_{eq.}\Big\{ \cos(\beta-\lambda)\left[\cos\bar{\varphi}\,\sin(-\psi) + \sin\bar{\varphi}\,\sin\gamma\,\cos(-\psi)\right] - \sin(\beta-\lambda)\cos\gamma\cos(-\psi)\Big\} \qquad 5)$$

$$r = \Omega_{eq.}\Big\{ \cos(\beta-\lambda)\left[-\cos\bar{\varphi}\,\cos(-\psi)\,\sin(-\theta) + \sin\bar{\varphi}\,\sin\gamma\,\sin(-\psi)\,\sin(-\theta) + \sin\bar{\varphi}\,\cos\gamma\,\cos(-\theta)\right] + \sin(\beta-\lambda)\left[-\cos\gamma\,\sin(-\psi)\,\sin(-\theta) + \sin\gamma\,\cos(-\theta)\right] \qquad 6)$$

When random values of $\bar{\alpha}$ and $\bar{\varphi}$ are to be used for Monte Carlo simulation of impact patterns, relations 2) and 3) will be used extensively. In simulation of impact patterns, symmetry considerations can also be employed such that only the left or right hand side of the pattern need be determined. The symmetry considerations are applied to the initial conditions by restricting the angle of attack to $0 < \bar{\alpha} \leq \frac{\pi}{2}$.

B. BODY-FIXED AXES

Ballistic trajectories of slowly-spinning rockets, missiles, bombs, and projectiles will usually be computed using the body-fixed axes option. For ballistic-type trajectories it will be convenient to select the horizontal component of the initial velocity in the direction of the positive X axis, in order that first quadrant values can be used for the Euler angles, θ and ψ.

Supplementary calculations will be required to determine the initial Euler angles if only the velocity vector and the angle of attack parameters, $\bar{\alpha}$ and ξ_i, are known or specified. The procedure involves finding the values of θ and ψ which satisfy the matrix equation

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} C\theta \cdot C\psi & C\theta \cdot S\psi & -S\theta \\ C\psi \cdot S\varphi \cdot S\theta - C\varphi \cdot S\psi & S\varphi \cdot S\theta \cdot S\psi + C\varphi \cdot C\psi & S\varphi \cdot C\theta \\ C\psi \cdot C\varphi \cdot S\theta + S\varphi \cdot S\psi & C\varphi \cdot S\theta \cdot S\psi - S\varphi \cdot C\psi & C\varphi \cdot C\theta \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}$$

subject to the constraints

$$\vec{\alpha} = \tan^{-1} \frac{\sqrt{v^2 + w^2}}{u}$$

$$\xi = \tan^{-1} \frac{w}{v}$$

$$\varphi_o = \text{CONSTANT}$$

For special cases, closed form solutions exist, i.e., for $\dot{Y} = \dot{Z} = \varphi_o = 0$

$$\tan \theta = \frac{\tan \vec{\alpha}}{\left[\frac{1}{\tan^2 \xi} + 1 \right]^{1/2}}$$

$$\cos \psi = \frac{\cos \theta}{\cos \vec{\alpha}}$$

More general velocity vector orientations with $\varphi_o \neq 0$ require complicated numerical solutions. Obviously, these problems do not exist if the Euler angles can be selected apriori. However, for multiple simulations, involving a fixed velocity vector, the above procedure will normally be required. A nearly general solution for θ , ψ , subject only to $\dot{Y} = 0$, i.e., $\tan \gamma = \dot{Z}/\dot{X}$, is given by the relations

$$\sin \theta \; [-\sin \gamma] + \cos \theta \; [\cos \gamma \; \cos \psi] = \cos \vec{\alpha}$$

$$\sin \theta \left[\frac{1}{\tan \psi} \right] + \cos \theta \left[\frac{\tan \gamma}{\sin \psi} \right] = \frac{\sin \varphi_o + \cos \varphi_o \tan \xi}{\sin \varphi_o \tan \xi - \cos \varphi_o}$$

C. INTEGRATION INTERVAL

For magnus rotor motion simulations, it will generally be advisable to select an integration time interval (TSTEP) such that about 100 integrations are performed for each cyclic period. For fixed-plane axes, the highest frequency will be the nutation frequency, while for body-fixed axes the highest frequency will correspond to the spin rate. Since the nutation frequency is approximately $p I_x/I$, a large saving in computing time can be achieved using the fixed-plane axes if $I_x/I \ll 1$.

When it is anticipated that only a short segment of a motion history will involve high frequencies, localized integration improvement can be achieved by using the re-correction option. A quaternion error parameter EPSMAX of about 1.0×10^{-7} would then be used in conjunction with a value of NCMAX of 3 or 4.

REFERENCES

1. Brunk, J. E., W. L. Davidson, R. W. Rakestraw, The Dynamics of Spinning Bodies at Large Angle of Attack, Air Force Office of Scientific Research Report No. AFOSR/DRA-62-3, January 1962, AD 275 437.

2. Murphy, Charles H., Free Flight Motion of Symmetric Missiles, BRL Report No. 1216, July 1963.

3. Tobak, Murray, Lewis B. Schiff and Victor L. Peterson, NASA Ames Research Center, Aerodynamics of Bodies of Revolution in Nonplanar Motion, AIAA Paper No. 68-20, AIAA 6th Aerospace Sciences Meeting, New York City, January 22-24, 1968.

4. Brunk, James E., Analytical Study of Bomblet Rotation Motions, Final Summary Report, Contract DA-18-108-AMC-236(A), for U. S. Army Edgewood Arsenal, Alpha Research, Inc. Report No. 66-236(A)-7, June 1966.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Alpha Research, Inc. Santa Barbara, California	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE
USER'S MANUAL: EXTENDED CAPABILITY MAGNUS ROTOR AND BALLISTIC BODY 6-DOF TRAJECTORY PROGRAM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report (6 October 1969 to 6 April 1970)

5. AUTHOR(S) (First name, middle initial, last name)
James E. Brunk

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1970	111	4

8a. CONTRACT OR GRANT NO. F08635-70-C-0012 b. PROJECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFATL-TR-70-40

10. DISTRIBUTION STATEMENT
This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Air Force Armament Laboratory (ATRA), Eglin Air Force Base, Florida 32542.

11. SUPPLEMENTARY NOTES AVAILABLE IN DDC	12. SPONSORING MILITARY ACTIVITY Air Force Armament Laboratory Air Force Systems Command Eglin Air Force Base, Florida 32542

13. ABSTRACT

A six-degrees-of-freedom trajectory program for quasi-symmetric rigid bodies is described. The equations of motion are developed such that either a body-fixed or fixed-plane moving coordinate system can be utilized. Provision is made for large angle and angular rate motions, such as are experienced by magnus rotor munitions.

The aerodynamic system permits the usual aeroballistic coefficients to be expressed as tabular functions of angle of attack and Mach number; in addition, the magnus force, magnus moment, and rolling moment coefficients can be tabular functions of the nondimensional spin parameter, pd/2V. Additional aerodynamic terms are provided to account for body-fixed aerodynamic asymmetries and/or control inputs, aerodynamic roll angle effects, flow asymmetry with respect to the angle of attack plane at zero spin, and lateral c. g. offset.

The computer program is written in General Fortran IV language compatible with CDC 6400, IBM 360, and GE 635 data processing machines. Included in the report are the program input and output formats, flow charts of the main program subroutines, and a complete program listing.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer programs						
Trajectories						
Aerodynamic characteristics						
Bomblets						